The computation of eigenvalues of large sparse matrices

Hawkins, Stuart C.

Award date:
1999

Awarding institution:
University of Bath

The computation of eigenvalues of large sparse matrices

submitted by

Stuart C. Hawkins

for the degree of Ph.D.

of the

University of Bath

1999

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Summary

Many of the most effective methods for computing eigenvalues of large sparse matrices are based on the shift-invert transformation. Applying this transformation involves solving a linear system with the shifted matrix. When this matrix is very large it is desirable to solve this system iteratively. This is not always appropriate; we show that the use of GMRES as an iterative solver in Inverse Iteration causes stagnation and we explain why. The Inverse Correction method does not stagnate. By exploiting the link between Inverse Correction and the Generalized Davidson method we provide an alternative view of the shift selection strategy in Inverse Correction.

The Jacobi-Davidson method is an extension of the Generalized Davidson method. Many variants of the Jacobi-Davidson method can be found in the literature. We show the superiority of one variant for generalised eigenvalue problems that arise in discretizations of the Navier Stokes equations. We show that this variant does not compute infinite eigenvalues, and we illustrate this numerically.

Implementing the Accelerated Rayleigh Quotient Iteration (ARQI) using iterative solvers such as GMRES is inefficient; convergence is slow because the shifted matrix is nearly singular. We show that it is possible to reformulate the solve so that small eigenvalues of the shifted matrix do not slow down GMRES. The new method obtained generalises the Jacobi-Davidson method. We show that the new method converges superlinearly, and in an important special case is equivalent to the ARQI and the Jacobi-Davidson method. The new method can be considerably cheaper to implement. We explain when this is the case and illustrate this analysis numerically.

Eigenvalues of preconditioned Jacobians are often easier to compute than those of the Jacobian. We develop a technique for approximating the eigenvalues of the Jacobian using those of a preconditioned Jacobian. The technique is applied to detect Hopf bifurcation.

Notation

In this thesis we have used the following notation:

x^H	The conjugate transpose of the vector x.
$\mathrm{mgs}(V)$	The orthonormal matrix whose columns are obtained by applying the modified Gram-Schmidt procedure to the columns of the matrix V.
$\langle x, y \rangle$	The inner product $x^H y$ of the vectors x and y.
$\mathcal{N}(A)$	The null space of the matrix A.
$\mathcal{R}(A)$	The range of the matrix A.
$\Lambda(A)$	The spectrum of the matrix A.
$\langle x_1, \ldots, x_n \rangle$	The span of the vectors $x_1, \ldots, x_n$.
$D(c, r)$	Disk centred at c with radius r.

In addition, we have used `typewriter style` text to represent matrices and vectors generated using Matlab instructions. For example, `diag(1:10)` represents the diagonal matrix whose diagonal has entries $1, 2, \ldots, 10$.

I would like to thank Alastair Spence—for supervising me during this project, and especially for all of the time and effort he has put into proofreading this thesis. Thanks also to the School of Mathematical Sciences at the University of Bath and the members of the Numerical Analysis Group. And to EPSRC whose support made this work possible.

I also wish to thank my parents and my brother for their constant support and encouragement.

I would like to thank all of my friends, especially Aaron, Gini and Paul—with special thanks to Andrew Falcon, Graeme Boswell, and Richard Cleyton who have been the greatest of friends.

Special thanks also to Karl for our excellent days off, and for his constant friendship. And to Suzy and Danielle for a thousand e-mails.

Contents

Chapter 1

Introduction

1.1 The eigenvalue problem

1.1.1 The standard eigenvalue problem

Let A be an $n \times n$ matrix. The *eigenvectors* and *eigenvalues* of A are nonzero vectors x and scalars λ which satisfy

$$Ax = \lambda x.$$

The need to compute eigenvalues and eigenvectors arises in applications such as vibration analysis, where eigenvalues of A correspond to resonant frequencies; and in stability analysis for dynamical systems, where the eigenvalues of Jacobian matrices divulge information about the stability of steady states.

We say that A is *diagonalisable* if there exists an $n \times n$ matrix X and a diagonal matrix Λ such that

$$AX = X\Lambda.$$

Clearly the ith column of X is an eigenvector of A whose corresponding eigenvalue is the ith entry in the diagonal of Λ.

For any A there exists a unitary $n \times n$ matrix S, and an upper triangular $n \times n$

matrix U, such that

$$AS = SU.$$

The columns of S are called *Schur vectors* of A—Schur vectors of A are not unique. It follows from the above that $S^H AS = U$. If A is Hermitian (or symmetric if A is real) then U is diagonal and the Schur vectors of A are eigenvectors.

If a nonzero vector y satisfies $y^T A = \lambda y^T$ then y is a *left eigenvector* of A. If λ is a simple eigenvalue of A with corresponding eigenvector x (properly a *right eigenvector*) and left eigenvector y, then we define the spectral projector for λ to be $\mathcal{P} = xy^T$.

Detailed discussion of the theory for the standard eigenvalue problem can be found in Golub and Van Loan [29], Saad [56], Wilkinson [75], and for the case of symmetric A, Parlett [50].

1.1.2 The generalised eigenvalue problem

Let B be an $n \times n$ matrix. The eigenvectors and "eigenvalues" of the *matrix pencil* (A, B) are nonzero vectors x and scalar pairs (α, β) which satisfy

$$\beta Ax = \alpha Bx.$$

For a given eigenvector x, the pair (α, β) is not unique—clearly any scalar multiple of (α, β) is also an "eigenvalue". If β is nonzero we will call the scalar $\lambda := \alpha/\beta$ an eigenvalue. Clearly

$$Ax = \lambda Bx.$$

If B is singular then "eigenvalues" $(\alpha, 0)$ will occur. We will refer to these as *infinite eigenvalues.*

When all of the eigenvalues of (A, B) are distinct there exist $n \times n$ matrices V and

W, and $n \times n$ diagonal matrices D_A and D_B, such that

$$W^H A V = D_A, \quad W^H B V = D_B.$$

The matrix W is the matrix of *left eigenvectors* of (A, B) and the matrix V is the matrix of *right eigenvectors* of (A, B).

For any (A, B) there exist $n \times n$ unitary matrices S_1 and S_2 and $n \times n$ upper triangular matrices R_A and R_B such that

$$S_1^H A S_2 = R_A, \quad S_1^H B S_2 = R_B.$$

This is an analogue of the Schur decomposition for the standard eigenvalue problem.

Detailed discussion of the theory for the generalised eigenvalue problem can be found in Saad [56], Stewart and Sun [72], and for the case of symmetric A and B, Parlett [50].

1.1.3 Small eigenvalue problems

Techniques for computing eigenvalues (and eigenvectors) of the matrix A fall into two broad categories:

(i) techniques for full matrices,

(ii) techniques for large, sparse, matrices.

The QR algorithm (Francis [25], [26], and Kublanovskaya [35]) computes Schur decompositions of small matrices. From a Schur decomposition the eigenvalues of A are immediately available, and their corresponding eigenvectors are easily computed.

The situation for the generalized eigenvalue problem is similar. The QZ algorithm (Moler and Stewart [41]) computes unitary matrices S_1 and S_2 such that $S_1^H A S_2$ and $S_1^H B S_2$ are upper triangular. From this decomposition the eigenvalues of (A, B) are immediately available, and as for the standard eigenvalue problem, their corresponding eigenvectors are easily computed.

1.2 Iterative methods

1.2.1 Large, sparse matrices

In this section we consider the standard eigenvalue problem $Ax = \lambda x$ where A is a large, sparse, matrix. When A is large, direct methods such as the QR method are often too expensive, and in any case, they compute all of the eigenvalues of A—often we are only interested in a small number of the eigenvalues. For example, stability analysis usually requires only the computation of the rightmost eigenvalue of A. In this section we discuss three methods which compute iteratively a sequence of approximations θ to an eigenvalue of A, (and/or approximations $\hat{x}$ to the corresponding eigenvector). A measure of the quality of an *approximate eigenpair* $(\hat{x}, \theta)$ is its *residual* $r := A\hat{x} - \theta\hat{x}$.

We assume that A is diagonalisable, and denote by $\lambda_1, \ldots, \lambda_n$ the eigenvalues of A, ordered (unless otherwise stated) by magnitude, so that

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|.$$

With this ordering we say that λ_1 is the *dominant* eigenvalue of A. We denote by $x_1, \ldots, x_n$ the eigenvectors corresponding to $\lambda_1, \ldots, \lambda_n$.

1.2.2 The Power method

The simplest iterative method is the *Power method.* If $|\lambda_1| > |\lambda_2|$ then, given a starting vector $\hat{x}^{(0)}$, the Power method computes a sequence of vectors $\hat{x}^{(1)}, \hat{x}^{(2)}, \ldots$ by repeated multiplications with A. This sequence converges to the eigenvector x_1 with convergence factor $|\lambda_2/\lambda_1|$.

An implementation of the Power method is given in Algorithm 1.1. The normalisation at line 1b may be done in several ways, commonly so that a particular component of $x^{(k)}$ is set to one. In any case, the ratio $\|y^{(k)}\|_2/\|\hat{x}^{(k)}\|_2$ converges to $|\lambda_1|$.

The Power method is discussed in more detail in Golub and Van Loan [29], Saad [56], Wilkinson [75], and for the case of symmetric A, Parlett [50].

Algorithm 1.1: The Power method

Choose initial guess vector $\hat{x}^{(0)}$.

1. For $k = 1, 2, \ldots$ do
   a) Compute $y^{(k)} = A\hat{x}^{(k-1)}$,
   b) Normalize, $\hat{x}^{(k)} = y^{(k)}/\|y^{(k)}\|_2$,
   c) Test for convergence of $\hat{x}^{(k)}$.

Algorithm 1.2: Subspace Iteration

Choose initial guess matrix $X^{(0)} = [\hat{x}_1^{(0)}, \ldots, \hat{x}_m^{(0)}]$.

1. For $k = 1, 2, \ldots$ do
   a) Compute $Y^{(k)} = AX^{(k-1)}$,
   b) Orthonormalize $Y^{(k)}$ into $X^{(k)}$,
   c) Test for convergence.

1.2.3 Subspace Iteration

Subspace Iteration is a development of the Power method which simultaneously computes approximations to a number of the Schur vectors of A. Thus Subspace Iteration is also known as *Simultaneous Iteration.* One can think of Subspace Iteration as the application of the Power method to m vectors simultaneously. In the Power method these would all converge to x_1, but if they are orthogonalised against one-another at each step then approximations to the first m Schur vectors are computed. Furthermore, the approximation to the Schur vector corresponding to the eigenvalue λ_i converges with convergence factor $|\lambda_{i+1}/\lambda_i|$ (see, for example, Saad [56, Ch.V]).

An implementation of Subspace Iteration is given in Algorithm 1.2.

1.2.4 The Rayleigh-Ritz procedure

The *Rayleigh-Ritz procedure* computes approximate eigenvectors of A which lie within a given subspace of $\mathbb{C}^n$, and is fundamental to many of the iterative eigenvalue solvers currently in use. To understand the Rayleigh-Ritz procedure we first define the *Rayleigh Quotient* of an approximate eigenvector $\hat{x}$.

Definition 1.1 *The Rayleigh Quotient of $\hat{x}$ is the scalar $\rho(\hat{x}) := \hat{x}^H A\hat{x}/\hat{x}^H\hat{x}$.*

Let $\mathcal{V}$ be a subspace of $\mathbb{C}^n$, with $\mathcal{V}$ spanned by the columns of the $n \times m$ orthonormal matrix V. If $\hat{x} \in \mathcal{V}$ is an approximate eigenvector of A, then there exists $y \in \mathbb{C}^m$ such

that $\hat{x} = Vy$. The approximate eigenpair, $\hat{x}$ with its Rayleigh-Quotient θ, has residual

$$\begin{aligned} r &:= A\hat{x} - \theta\hat{x} \\ &= AVy - \theta Vy, \end{aligned}$$

and we see that

$$\begin{aligned} V^H r &= V^H AVy - \theta V^H Vy \\ &= Hy - \theta y, \end{aligned}$$

where H is the $m \times m$ matrix $V^H AV$. Working backwards we see that if (y, θ) is an eigenpair of H then $(\hat{x}, \theta) = (Vy, \theta)$ is an approximate eigenpair of A, whose residual is *orthogonal* to $\mathcal{V}$.

By computing eigenvectors of H the Rayleigh-Ritz procedure extracts approximate eigenvectors of A from the subspace $\mathcal{V}$. To do this requires the solution of an $m \times m$ eigenproblem but for low dimensional subspaces $\mathcal{V}$ this eigenvalue problem will be small. Small eigenvalue problems can be efficiently solved using the QR method (see Section 1.1.3).

Approximate eigenvectors $\hat{x}$ computed by the Rayleigh-Ritz procedure are called Ritz vectors and their corresponding approximate eigenvalues are called Ritz values. The pair $(\hat{x}, \theta)$ is called a Ritz-pair.

A detailed discussion of the Rayleigh-Ritz procedure can be found in Saad [56], and for the case of symmetric A, Parlett [50].

1.2.5 Accelerated Subspace Iteration

Recall that Subspace Iteration computes a matrix $X^{(k)}$ whose columns approximate Schur vectors of A. One would expect the subspace $\langle \hat{x}_1^{(k)}, \ldots, \hat{x}_m^{(k)} \rangle$ to contain good approximations to eigenvectors of A. Applying the Rayleigh-Ritz procedure to this space in order to compute these approximate eigenvectors leads to Accelerated Subspace Iteration (Algorithm 1.3).

Algorithm 1.3: Accelerated Subspace Iteration

Choose initial guess matrix $X^{(0)} = [\hat{x}_1^{(0)}, \ldots, \hat{x}_m^{(0)}]$.

1. For $k = 1, 2, \ldots$ do
   a) Compute $Y^{(k)} = AX^{(k-1)}$,
   b) Orthonormalize $Y^{(k)}$ into $X^{(k)}$,
   c) Compute approximate eigenvalues and eigenvectors of A using the Rayleigh-Ritz procedure.
   d) Test for convergence.

The convergence rate of Accelerated Subspace Iteration is quantified by the following theorem, which is due to Stewart [71].

Theorem 1.2 (Stewart [71])

Algorithm 1.3 computes at step k an approximation θ_i to λ_i, $(i = 1, \ldots, m)$ with

$$|\theta_i - \lambda_i| = \mathcal{O}\left(\left|\frac{\lambda_{m+1}}{\lambda_i}\right| + \epsilon_k\right)^k$$

where ϵ_k tends to zero as k tends to infinity.

1.2.6 Arnoldi's Method

The sequence of vectors $\hat{x}^{(0)}, \hat{x}^{(1)}, \ldots$ computed by the Power method converges to an eigenvector of A. Arnoldi's method (Saad [54]) uses the Rayleigh-Ritz method to compute approximate eigenvectors of A from the subspace spanned by this sequence of vectors. The space spanned by a finite number of these vectors is called a Krylov subspace.

Definition 1.3 *The Krylov subspace of dimension k, generated by A and v, is the space*

$$\mathcal{K}_k(A, v) := \langle v, Av, \ldots, A^{k-1}v \rangle.$$

Arnoldi's method at step k extracts approximate eigenvectors of A from the Krylov subspace $\mathcal{K}_k(A, v)$. An implementation of Arnoldi's method is given in Algorithm 1.4.

The potential convergence rate of Arnoldi's method is illustrated by the following theorem.

Algorithm 1.4: Arnoldi's method

Choose initial guess vector $v^{(1)}$,

1. For $j = 1, 2, \ldots, m$ do
   a) Let $w^{(j)} = Av^{(j)}$,
   b) For $i = 1, 2, \ldots, j$
      i) $h_{ij} = \langle v^{(i)}, w^{(j)} \rangle$,
      ii) $w^{(j)} = w^{(j)} - h_{ij} v^{(i)}$,
   c) Let $h_{j+1,j} = \|w^{(j)}\|_2$, and $v^{(j+1)} = w^{(j)}/h_{j+1,j}$,
2. Compute the eigenvalues θ and eigenvectors y of the matrix $H_m := [h_{ij}]_{i,j=1}^{m}$, where $h_{ij} := 0$ for $i > j + 1$.
3. Test for convergence.

Theorem 1.4 (Saad [56, Proposition 6.10])

Suppose that the n eigenvalues of A are simple and that $\lambda_2, \ldots, \lambda_n$ are enclosed in a circle centred at ξ and passing through λ_2, without enclosing λ_1. Then the approximation θ to λ_1, computed at step k of Arnoldi's method, satisfies

$$|\theta - \lambda_1| \leq c \left| \frac{\lambda_2 - \xi}{\lambda_1 - \xi} \right|^{k-1}$$

for some constant c.

This bound on the convergence rate assumes that the eigenvalues $\lambda_2, \ldots, \lambda_n$ lie within a circle. The convergence rate observed in practise will depend very much on the distribution of the eigenvalues of A. Convergence estimates for some other eigenvalue distributions are given in Saad [56].

Remarks

(i) The matrix H_j in Arnoldi's method is *upper Hessenberg*. This reduces the cost of computing its eigenvalues using the QR method.

(ii) If A is symmetric then H_j is *tridiagonal* and could have been computed using the three term recurrence of the Lanczos method (see Parlett [50]). In fact Arnoldi's method reduces to the (symmetric) Lanczos method when A is symmetric.

Arnoldi's method at step k computes an $n \times (k+1)$ orthonormal matrix $V_{k+1} = [v^{(1)}, \ldots, v^{(k+1)}]$ and a $(k+1) \times k$ upper Hessenberg matrix $H_{k+1,k} = [h_{i,j}]$, with

$$AV_k = V_{k+1} H_{k+1,k}.$$

This equation may be rewritten

$$AV_k = V_k H_k + h_{k+1,k} v^{(k+1)} e_k^T \tag{1.1}$$

where H_k is the $k \times k$ matrix obtained from $H_{k+1,k}$ by removing its last row. Equation (1.1) is known as the Arnoldi factorisation.

From the Arnoldi factorisation we have the following lemma.

Lemma 1.5

The residuals of Ritz pairs obtained from a Krylov subspace generated by A are in the same direction.

Proof Suppose that $(\hat{x}_1, \theta_1)$ and $(\hat{x}_2, \theta_2)$ are Ritz pairs of A generated from a Krylov subspace of dimension k generated by A and some starting vector $v^{(1)}$. Then there exist vectors $y_1, y_2 \in \mathbb{C}^k$ and an $n \times k$ orthonormal matrix V such that $\hat{x}_1 = V y_1$, $\hat{x}_2 = V y_2$.

The residual of $(\hat{x}_1, \theta_1)$ is

$$\begin{aligned} r_1 &= A\hat{x}_1 - \theta_1 \hat{x}_1 \\ &= AVy_1 - \theta_1 V y_1 \\ &= (VH_k + h_{k+1,k} v^{(k+1)} e_k^T) y_1 - \theta_1 V y_1 \\ &= V\theta_1 y_1 + h_{k+1,k} v^{(k+1)} e_k^T y_1 - \theta_1 V y_1 \\ &= v^{(k+1)} (h_{k+1,k} e_k^T y_1). \end{aligned}$$

Similarly $r_2 = v^{(k+1)} (h_{k+1,k} e_k^T y_2)$. Both r_1 and r_2 lie in the direction of $v^{(k+1)}$. □

1.3 Matrix transformations

1.3.1 Matrix transformations

In the previous section we discussed three iterative methods for computing eigenvalues and eigenvectors of a given (large sparse) matrix A. In each case the rate of convergence is in some way dependent on the eigenvalue distribution of A, and is usually related to the *separation* of the desired eigenvalue from one, or all, of the remaining eigenvalues.

Increased rates of convergence can be obtained by applying the iterative methods of the previous section to matrix transformations of A which preserve A's eigenvectors, but change its eigenvalues.

In addition, although the iterative methods of the previous section cannot be applied directly for the generalised eigenvalue problem, they can be applied to some transformations of the generalised eigenvalue problem.

1.3.2 The Shift-Invert transformation

The *Shift-Invert* transformation

$$T_{SI} := (A - sB)^{-1}B,$$

with *shift* s, transforms eigenvalues of A that are close to s to large eigenvalues of T_{SI}. It is clear that if λ is an eigenvalue of (A, B), then $1/(\lambda - s)$ is an eigenvalue of T_{SI}, and that these eigenvalues of (A, B) and T_{SI} respectively share the same eigenvector.

The Power method applied to the matrix T_{SI} is called the *Inverse Power* method, or *Inverse Iteration.* Inverse Iteration for the standard eigenvalue problem is discussed in Golub and Van Loan [29], Saad [56], Wilkinson [75], and in the case of symmetric A, in Parlett [50]. Inverse Iteration for the generalised eigenvalue problem is discussed in Saad [56] and in the case of symmetric A and B, Parlett [50]. An implementation of Inverse Iteration is given in Algorithm 1.5.

Special mention should be made of the *Rayleigh Quotient Iteration* (RQI), which at each step uses the Rayleigh Quotient of the current approximate eigenvector for the shift

Algorithm 1.5: Inverse Iteration

Choose initial guess vector $\hat{x}^{(0)}$.

1. For $k = 1, 2, \ldots$ do
   a) Solve $(A - sI)y^{(k)} = \hat{x}^{(k-1)}$,
   b) Normalize, $\hat{x}^{(k)} = y^{(k)}/\|y^{(k)}\|_2$,
   c) Test for convergence of $\hat{x}^{(k)}$.

Algorithm 1.6: Rayleigh Quotient Iteration

Choose initial guess vector $\hat{x}^{(0)}$.

1. For $k = 1, 2, \ldots$ do
   a) Compute $\rho^{(k-1)} = \rho(\hat{x}^{(k-1)})$,
   b) Solve $(A - \rho^{(k-1)}I)y^{(k)} = \hat{x}^{(k-1)}$,
   c) Normalize, $\hat{x}^{(k)} = y^{(k)}/\|y^{(k)}\|_2$,
   d) Test for convergence of $\hat{x}^{(k)}$.

in the Shift-Invert transformation. For symmetric A this leads to cubic convergence. A full discussion is given in Ostrowski [47, 48]. Generalisations of the Rayleigh Quotient Iteration for non-normal standard eigenvalue problems are given in Parlett [49]. A technique for combining Inverse Iteration and the Rayleigh Quotient Iteration, thereby preventing convergence to the *wrong* eigenvalue, is given in Szyld [73].

Subspace Iteration and Arnoldi's method (or the Lanczos method) can also be applied to T_{SI}, see Scott [62], Ericsson [22], Ericsson and Ruhe [23], and Omid, Parlett, Ericsson, and Jensen [45].

1.3.3 The Cayley transform

The *Cayley transform*

$$T_C := (A - \sigma B)^{-1}(A - sB)$$

transforms eigenvalues λ of (A, B) to eigenvalues $\theta = (\lambda - \sigma)^{-1}(\lambda - s)$ of T_C. In particular, lines $L(a) := \{a + bi : b \in \mathbb{R}\}$ map to circles under the Cayley transform, and the line $L(\frac{1}{2}(\sigma + s))$ maps to the unit circle. We apply the Cayley transform and discuss its mapping properties more fully in Section 2.2.4.

For a discussion of the Cayley transform see Garratt [28], and Meerbergen [39].

1.3.4 Chebyshev polynomials

For completeness we mention the Chebyshev polynomial, although we do not make use of it in this thesis. Suppose that we wish to compute the eigenvalue λ_1 of the matrix A. Then it is possible to construct a Chebyshev polynomial p such that $p(\lambda_1)$ is large compared with $p(\lambda_i)$, for $i \neq 1$. Iterative methods applied to $p(A)$ will compute λ_1 easily. This technique is often used in Subspace Iteration and, in various formulations, Arnoldi's method—see Saad [57], and for an overview, Meerberen [39].

1.4 Other iterative methods

1.4.1 Restarting

At the kth step of Arnoldi's method the storage of a sequence of k vectors is required, a newly computed vector must be orthogonalised against k other vectors, and the eigenpairs of a $k \times k$ matrix must be computed. Consequently the cost of a step of Arnoldi's method rises with the size of the subspace.

There comes a point at which the cost of continuing with Arnoldi's method is too high. To reduce the cost of continuing one may at this point *restart* the iteration, that is, start a new Arnoldi iteration with a new initial guess vector. The new initial guess vector will be chosen from the current Krylov subspace—a common choice is the current approximation to some desired eigenvector. Whatever the choice, one may think of it as the application of some polynomial ψ of A to the original initial guess vector $v^{(1)}$. Thus the new starting vector is $\psi(A)v^{(1)}$. It is easy to see that we might construct ψ to *damp* components in some unwanted eigenvalues of A, for example, we might construct an appropriate Chebyshev polynomial.

Sorensen [67] shows that this polynomial restarting may be applied implicitly using a shifted QR iteration. This leads to the *Implicitly Restarted Arnoldi* method [68].

1.4.2 Bisection method

The bisection method is usually used to compute regions within which eigenvalues of the symmetric matrix A lie, but can be used to compute accurate approximations to

eigenvalues. This method relies upon the *triangular factorisation* $A - \sigma I = L\Delta L^H$ (see Parlett [50]) where L is lower triangular, and Δ is diagonal. The number of negative entries on the diagonal of Δ is equal to the number of negative eigenvalues of $A - \sigma I$, that is, the number of eigenvalues of A that lie to the left of σ. An adaptation of the bisection method in which it is combined with the Rayleigh Quotient Iteration is given in Scott [63].

1.4.3 Davidson's method

Davidson's method [16] is closely related to the Lanczos method applied to the preconditioned matrix $(D-\theta I)^{-1}(A-\theta I)$ with shift θ, and where D is the diagonal of A. The difference between Davidson's method and the preconditioned Lanczos method is that θ at each step is the Rayleigh Quotient of the current approximate eigenvector—so θ is not constant. Generalizations of Davidson's method replace D by some other easily inverted "approximation" to A, see Morgan [42] and Morgan and Scott [43]. We discuss the Generalized Davidson method in more detail in Chapter 2.

An alternative preconditioning technique for the Lanczos method is described in Morgan and Scott [44].

1.4.4 The Jacobi-Davidson method

The Jacobi-Davidson method (see Olsen, Jørgensen and Simons [46], Sleijpen and Van der Vorst [66], and Stathopoulos, Saad and Fischer [69]) is an adaptation of Davidson's method which expands the subspace with a vector computed to be orthogonal to the current approximate eigenvector. We discuss this method in detail in Chapter 2.

1.4.5 The Rational Krylov method

The Rational Krylov method of Ruhe [52], [53], is similar to the Shift-Invert Arnoldi method—but the shift used in the shift-invert transformation is not constant. Great flexibility is allowed in the choice of shift, and the vector to which the shifted and inverted matrix is applied may be *any* vector in the computed subspace.

1.5 Overview

A brief overview of the remaining chapters of this thesis is as follows.

We begin Chapter 2 by showing that it is unwise to use GMRES to solve the linear systems arising in Inverse Iteration. In the remainder of the chapter we discuss two methods that are based upon the Shift-Invert transformation and that tolerate solving of the linear systems iteratively. The first such method is the Inverse Correction method. We show that the Inverse Correction method is related to the Generalized Davidson method and give an alternative view of the shift selection strategy. The second such method is the Jacobi-Davidson method. We consider the application of the Jacobi-Davidson method to generalized eigenvalue problems that have a particular block structure that arises in mixed finite element discretizations of the Stokes and Navier Stokes equations. The Jacobi-Davidson method exists in many forms—there is no clear best form but we prove that one in particular does not compute spurious eigenvalues for this problem.

In Chapter 3 we discuss in detail the GMRES algorithm, and focus on polynomial based convergence results relating the convergence rate of GMRES to eigenvalue distribution. Results giving improved bounds on the norm of the residual vector for matrices with outlying eigenvalues are extended to estimate the gain in "removing" outlying eigenvalues that are close to the origin. A deflation technique similar to Wielandt deflation produces matrices which effectively have their outlying eigenvalues removed. In the main result of Chapter 3 we show that GMRES can work more quickly for such matrices.

In Chapter 4 we consider replacing the linear systems occurring in Shift-Invert methods with projected systems of the type discussed in Chapter 3. We apply the ideas developed to the Rayleigh Quotient Iteration (RQI) to produce a new method which implements the RQI using GMRES in place of direct solvers. The new method is cheaper to implement than the standard RQI implemented using GMRES.

In Chapter 5 we derive from the Accelerated Rayleigh Quotient Iteration (ARQI) a new method called the Iterative Accelerated Rayleigh Quotient Iteration (IARQI) which generalises the Jacobi-Davidson method. We state conditions under which the

Iterative Accelerated Rayleigh Quotient Iteration is mathematically equivalent to the Accelerated Rayleigh Quotient Iteration. The Iterative Accelerated Rayleigh Quotient Iteration uses GMRES to solve the linear systems which arise at each step, and which involve deflated matrices of the type discussed in Chapter 3. These systems can be solved more cheaply than those arising in the ARQI and Jacobi-Davidson methods. Numerical results are given for a number of test matrices.

In stability analysis the rightmost eigenvalues of Jacobian matrices divulge information about steady states. For some semilinear PDEs we find that the eigenvalues of a preconditioned Jacobian are cheaply available, but are not usually useful unless it is possible to obtain estimates of the eigenvalues of the Jacobian from them. In Chapter 6 we show that for certain preconditioners it is possible to correct the eigenvalues of the preconditioned Jacobian to obtain second order approximations to the eigenvalues of the Jacobian itself. We then present a method for accurately approximating bifurcation points which needs neither the eigenvalues of the Jacobian or approximations to them.

Chapter 2

Iterative solvers to compute eigenvalues

2.1 Introduction

Methods based on the shift-invert transformation take advantage of the improved separation of some of the eigenvalues of the shifted, inverted matrix. The accurate solution of a number of linear systems is necessary in these methods, and is often performed using a direct solver. Solving large sparse linear systems using a direct solver is usually expensive. We consider methods which are based on the shift-invert transformation, and for which it is appropriate to use iterative solvers to solve the linear systems.

We begin with a warning to those who would implement Inverse Iteration using GMRES as a solver. We show in Section 2.2.1 that injudicious shift selection will cause stagnation of Inverse Iteration. Reformulating the linear system to be solved in Inverse Iteration leads to the Inverse Correction method (Rüde and Schmid [51]) which does not suffer from this stagnation effect. In Section 2.2.3 we show that Inverse Correction is closely related to the Generalized Davidson method (Morgan [42] and Morgan and Scott [43]). Taking advantage of this relation we present in Section 2.2.6 an alternative analysis of the part played by the shift in the convergence of the Inverse Correction method.

Perversely, solving too accurately in the Generalized Davidson method (GD) can

cause stagnation. A modification of GD which ensures that the correction computed is orthogonal to the current approximate eigenvector leads to the Jacobi-Davidson method [6], [7], [8], [9], [18], [24], [64], [66]. In Section 2.3 we discuss some of the many formulations of Jacobi-Davidson. In Section 2.3.4 we discuss Jacobi-Davidson for the generalized eigenvalue problem and show the superiority of one of its variants for block generalized eigenvalue problems that arise in mixed finite element discretizations of the Stokes and Navier-Stokes equations.

An outline for this chapter is as follows. In Section 2.2.1 we show that implementing Inverse Iteration with iterative solvers is unreliable. In Section 2.2.2 we introduce the Inverse Correction method and in the remainder of Section 2.2 we show how the shift-selection strategy alters the convergence rate.

In Section 2.3 we give a brief description of the Jacobi-Davidson method for the standard eigenvalue problem, including three different formulations of the correction equation in Section 2.3.2, and the incorporation of deflation in Section 2.3.3. In Section 2.3.4 we discuss the application of Jacobi-Davidson to the generalised eigenvalue problem, and describe some of the variants available. We show how preconditioning may be used for Jacobi-Davidson in Section 2.3.5, and for deflated Jacobi-Davidson in Section 2.3.6.

In Section 2.4 we discuss a particular generalised eigenvalue problem that has special block structure. This block structure arises in mixed-finite elements discretizations of the Stokes and Navier Stokes equations. We show that one particular variant of Jacobi-Davidson is superior to the others in that it does not compute spurious eigenvalues.

2.2 Inverse Correction

2.2.1 Iterative solves in Inverse Iteration

Recall Inverse Iteration (Algorithm 1.5) and the Rayleigh Quotient Iteration (RQI) (Algorithm 1.6). These methods compute a single eigenvalue of a matrix, and its corresponding eigenvector. At each step of these methods a linear system involving the shifted matrix must be solved. Direct methods for solving this system are often used,

Algorithm 2.1: Inverse Iteration (Schmid)

Choose initial guess vector $\hat{x}^{(0)}$ and scalar σ.

1. For $k = 1, 2, \ldots$ do
   a) Compute $\rho^{(k-1)} = \rho(\hat{x}^{(k-1)})$ and $\widehat{\rho}^{(k-1)} = \sigma\rho^{(k-1)}$,
   b) Solve $(A - \widehat{\rho}^{(k-1)}I)y^{(k)} = \hat{x}^{(k-1)}$,
   c) Normalize, $\hat{x}^{(k)} = y^{(k)}/\|y^{(k)}\|_2$,
   d) Test for convergence of $\hat{x}^{(k)}$.

particularly in Inverse Iteration where an LU factorisation may be computed at the start of the iteration and reused. Iterative methods are alternatives to direct methods which, when the matrix is large and sparse, are often cheaper.

We now discuss Algorithm 2.1 (Rüde and Schmid [51]) which is a variant of Inverse Iteration, and is closely related to the Rayleigh Quotient Iteration. This algorithm differs from the Rayleigh Quotient Iteration in the shift used at each step. In this algorithm the shift is some scalar multiple of the Rayleigh Quotient of the current approximate eigenvector. In the RQI the shift is simply the Rayleigh Quotient. When an approximate eigenvector is close to convergence its Rayleigh Quotient will be very close to an eigenvalue and the shifted matrix in the Rayleigh Quotient Iteration will be nearly singular. Iterative solvers will work more slowly in this case. Choosing a shift which is some small perturbation of the Rayleigh Quotient will, loosely speaking, make the shifted matrix less singular. Note that Algorithm 2.1 generalises the Rayleigh Quotient Iteration.

The following example illustrates that care must be taken in applying Algorithm 2.1 with iterative solvers such as GMRES (Saad [59], also Chapter 3).

Example 2.1

In this example A is a 20×20 matrix with eigenvalues $-18, -17, \ldots, -1$ and $0.5, 1$. We apply Algorithm 2.1 with $\sigma = 1.05$ to compute the rightmost eigenvalue 1 of A. We use GMRES with zero initial guess vector to solve the linear system at each step.

Figure 2-1 shows the residual norm plotted against number of iterations for four applications of Algorithm 2.1, implemented with GMRES where the linear systems $Ay = b$ are solved to tolerance tol $:= \|b - Ay\|_2/\|b\|_2 = 1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}, 1 \times$

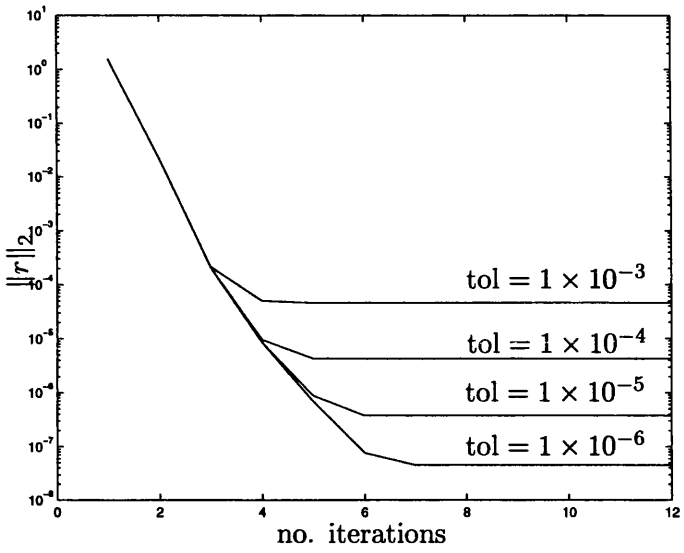


Figure 2-1: Residual norm plotted against number of iterations for Example 2.1.

10^{-6}. In the figure we see that in each case the residual norm stagnates. The value of the residual norm at which the method stagnates decreases with tol.

The behaviour in this example is explained in the following lemma.

Lemma 2.1

Let the unit vector $\hat{x}$ be an approximate eigenvector of A with Rayleigh Quotient θ and residual $r = A\hat{x} - \theta\hat{x}$. Then the approximate solution $\alpha\hat{x}$ in $\langle\hat{x}\rangle$, which minimises the residual norm for the system

$$(A - (\theta + \epsilon)I)y = \hat{x}, \tag{2.1}$$

is given by $\alpha = -\epsilon/(\epsilon^2 + \|r\|_2^2)$. This has residual norm $\|r\|_2/\sqrt{\|r\|_2^2 + \epsilon^2}$ as an approximate solution of (2.1).

Proof

Let $\alpha\hat{x}$ be the vector in $\langle\hat{x}\rangle$ minimising the residual for the system

$$(A - (\theta + \epsilon)I)y = \hat{x}.$$

The residual of $\alpha\hat{x}$ is

$$\begin{aligned} & \hat{x} - (A - (\theta + \epsilon)I)\alpha\hat{x} \\ = \; & \hat{x} - (A - \theta I)\alpha\hat{x} + \epsilon\alpha\hat{x} \\ = \; & (1 + \epsilon\alpha)\hat{x} - \alpha r. \end{aligned}$$

Taking norms and observing that $r \perp \hat{x}$ we see that the norm of this is minimised for $\alpha = -\epsilon\|\hat{x}\|_2^2/(\epsilon^2\|\hat{x}\|_2^2 + \|r\|_2^2)$. □

In Algorithm 2.1 we wish to solve

$$(A - \sigma\theta I)y = \hat{x} \tag{2.2}$$

using GMRES. Commonly the initial guess vector required by GMRES is taken to be the zero vector and we assume that this is the case. Writing $\sigma = 1 + \epsilon/\theta$ we see that (2.2) is of the form (2.1). By Lemma 2.1 the approximate solution at the first step of GMRES is $\alpha\hat{x}$ where $\alpha = -\epsilon/(\epsilon^2 + \|r\|_2^2)$. This approximate solution has residual $\|r\|_2/\sqrt{\|r\|_2^2 + \epsilon^2}$. Clearly, if

$$\|r\|_2^2 < \frac{\mathrm{tol}^2\epsilon^2}{1 - \mathrm{tol}^2}$$

then $\alpha\hat{x}$ will be deemed a satisfactory approximate solution of (2.2) and the iteration stagnates.

Recall that in Example 2.1 stagnation of Algorithm 2.1 was observed for $\epsilon = 0.05$. If the solves are performed in GMRES to tolerance $\mathrm{tol} = 1 \times 10^{-\alpha}$ then we expect stagnation to occur when the residual norm of $\hat{x}$ drops below

$$\begin{aligned} & \left(\frac{\mathrm{tol}^2\epsilon^2}{1 - \mathrm{tol}^2}\right)^{\frac{1}{2}} \\ \approx \; & \left(1 \times 10^{-2\alpha} \cdot 25 \times 10^{-4}\right)^{\frac{1}{2}} \\ \approx \; & 5 \times 10^{-(2+\alpha)}, \\ \approx \; & 10^{-(2+\alpha-0.7)}. \end{aligned}$$

Algorithm 2.2: Inverse Correction

Choose initial guess vector $\hat{x}^{(0)}$ and scalar σ.

1. For $k = 1, 2, \ldots$ do
   a) Compute $\rho^{(k-1)} = \rho(\hat{x}^{(k-1)})$, $\widehat{\rho}^{(k-1)} = \sigma\rho^{(k-1)}$, and $r = (A - \rho^{(k-1)}I)\hat{x}^{(k-1)}$,
   b) Solve $(A - \widehat{\rho}^{(k-1)}I)z^{(k)} = r$,
   c) Compute $\hat{x}^{(k)} = \hat{x}^{(k-1)} - z^{(k)}$
   d) Test for convergence of $\hat{x}^{(k)}$.

Clearly the results in Example 2.1 agree with the theory.

Remark

If $\epsilon = 0$ then $\alpha = 0$, and $\alpha\hat{x}$ has residual norm 1 as a solution of (2.2).

This analysis indicates why problems are likely to occur when, as in Example 2.1, we apply Algorithm 2.1 using GMRES.

The implementation of GMRES for the general system $Ax = b$ is discussed in detail in Section 3.2.3.

2.2.2 Inverse Correction

To combat stagnation of Inverse Iteration when implemented with GMRES, Rüde and Schmid [51] propose the Inverse Correction method. Although it is mathematically equivalent to Inverse Iteration, in Inverse Correction a scalar multiple of the solution y of system (2.1) is computed by adding to $\hat{x}$ a correction z which is obtained by solving

$$(A - \omega I)z = r.$$

Here $r = (A - \theta I)\hat{x}$, the residual of the eigenpair $(\hat{x}, \theta)$ of A. We require $\omega \neq \theta$ (otherwise $\hat{x} = z$). The new approximate eigenvector is given by $\hat{x} - z$, and writing

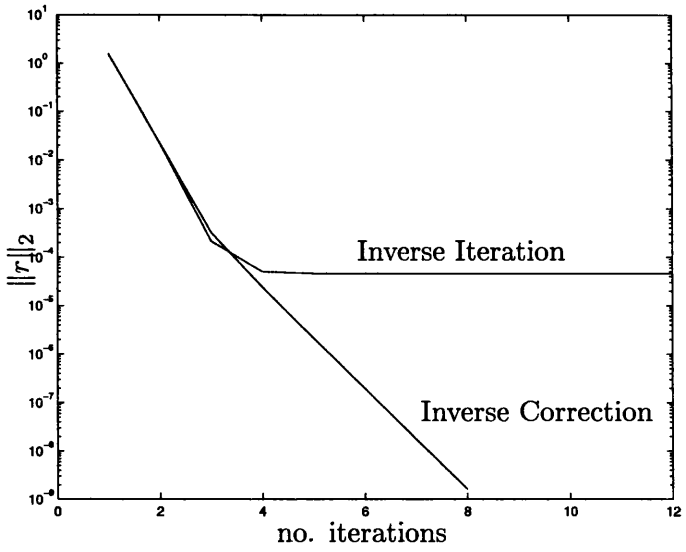


Figure 2-2: Residual norm plotted against number of iterations for Inverse Correction and Inverse Iteration applied to the matrix in Example 2.1. Solves are performed with tol $= 1 \times 10^{-3}$.

$\epsilon = \omega - \theta$ we have

$$\begin{aligned}
\hat{x} - z &= \hat{x} - (A - \omega I)^{-1} r \\
&= \hat{x} - (A - \omega I)^{-1}(A - \theta I)\hat{x} \\
&= \hat{x} - (A - \omega I)^{-1}(A - (\theta + \epsilon)I + \epsilon I)\hat{x} \\
&= \hat{x} - \hat{x} - \epsilon(A - \omega I)^{-1}\hat{x} \\
&= -\epsilon(A - \omega I)^{-1}\hat{x} \\
&= -\epsilon y.
\end{aligned}$$

An implementation of Inverse Correction is given in Algorithm 2.2. To illustrate the improvement over Inverse Iteration, results for Inverse Correction applied to the matrix in Example 2.1 are shown in Figure 2-2. Rüde and Schmid give a brief analysis of the behaviour of Inverse Correction which we now summarise.

Suppose that A is diagonalisable with eigenvalues $\lambda_1, \ldots, \lambda_n$ and corresponding eigenvectors $x_1, \ldots, x_n$. For convenience we assume that we are trying to compute λ_1 and x_1.

Let $\hat{x}^{(k)} \approx x_1$ and write

$$\hat{x}^{(k)} = \sum_{i=1}^{n} \alpha_i x_i$$

for some scalars $\alpha_1, \ldots, \alpha_n$. Then, solving exactly, the new approximate eigenvector may be written

$$\hat{x}^{(k+1)} = \sum_{i=1}^{n} \sigma_i \alpha_i x_i$$

where each

$$\sigma_i = \frac{\lambda_1(\tau - \hat{\tau})}{\lambda_i - \lambda_1(1 + \hat{\tau})}$$

with τ satisfying $(1 + \tau)\lambda_1 = \theta$, $\hat{\tau}$ satisfying $(1 + \hat{\tau})\lambda_1 = \omega$.

In particular we have

$$\begin{aligned} \sigma_1 &= \frac{\lambda_1(\tau - \hat{\tau})}{-\lambda_1 \hat{\tau}} \\ &= \frac{\hat{\tau} - \tau}{\hat{\tau}}. \end{aligned}$$

Setting $\hat{\tau} = \gamma\tau$ gives $\sigma_1 = (\gamma - 1)/\gamma$. Rüde and Schmid propose to make $\sigma_1 = \mathcal{O}(1)$ by choosing γ large.

The "amplification factor" for the ith eigenvector is

$$\sigma_i = \frac{\lambda_1(\tau - \hat{\tau})}{\lambda_i - \lambda_1(1 + \tau)}.$$

Rüde and Schmid propose to make σ_i small by minimising $\tau - \hat{\tau}$.

To achieve the above aims, that is to have $\sigma_1 = \mathcal{O}(1)$ and $\sigma_i \approx 0$ $(i = 2, \ldots, n)$, Rüde and Schmid recommend making θ as close to the eigenvalue λ_1 as possible, and taking ω as small a perturbation of θ as possible.

2.2.3 Inverse Correction and Davidson's method

In this section let $\hat{x}$ denote an approximate eigenvector with Rayleigh Quotient θ and residual $r := A\hat{x} - \theta\hat{x}$. Davidson's method for symmetric A [16] computes at each step a correction

$$z = (D - \omega I)^{-1} r,$$

where $D = \text{diag}(A)$. The Generalized Davidson method [42],[43] computes at each step a correction

$$z = (M - \omega I)^{-1} r,$$

where M is some easily inverted approximation to A. Here A need not be symmetric. Usually $\omega = \theta$. This correction is the same as the correction computed in Inverse Correction. The Generalized Davidson Method (GD) differs from Inverse Correction in the way that z is used—in GD z is used to expand a subspace, from which approximate eigenvectors can be computed, using for example the Rayleigh-Ritz procedure.

Crouzeix, Philippe, and Sadkane [14] and Sadkane [60] give convergence results for Davidson's method. Morgan [42] and Morgan and Scott [43] suggest studying the Generalized Davidson method by looking at the spectrum of the matrix $(M-\theta I)^{-1}(A-\theta I)$, which is closely related to the Cayley Transform. We use this approach to study the convergence of the Inverse Correction method.

2.2.4 The Cayley Transform

The *Cayley transform* (see Section 1.3.3) of the matrix A is the matrix

$$T_C = (A - \omega I)^{-1}(A - \theta I),$$

for scalar shifts ω and θ. Each eigenvalue λ_i of A maps to a corresponding eigenvalue

$$\eta_i = \frac{\lambda_i - \theta}{\lambda_i - \omega}$$

of T_C. The eigenvectors are unchanged. The properties of the Cayley transform are discussed in detail in Garratt [28, Ch.2].

We now present two approaches to the selection of ω and θ:

1. Attempt to make η_1 small compared with the other eigenvalues of T_C.

2. Attempt to make $\eta_2, \ldots, \eta_n$ as close to 1 as possible.

The first approach appeals to tradition, in that the separation of η_1 from the remainder of the spectrum is maximised. We show that the second approach also has its merits.

Approach 1 Clearly when θ is close to λ_1 we have $\eta_1 \approx 0$. We assume this is the case and examine the ratio

$$\begin{aligned}
\eta_1/\eta_i &= \left(\frac{\lambda_1 - \theta}{\lambda_1 - \omega}\right) / \left(\frac{\lambda_i - \theta}{\lambda_i - \omega}\right) \\
&= \left(\frac{\lambda_1 - \theta}{\lambda_i - \theta}\right) \left(\frac{\lambda_i - \omega}{\lambda_1 - \omega}\right) \\
&= \left(\frac{\lambda_1 - \theta}{\lambda_i - \theta}\right) \left(\frac{\lambda_1 - \omega + \lambda_i - \lambda_1}{\lambda_1 - \omega}\right) \\
&= \left(\frac{\lambda_1 - \theta}{\lambda_i - \theta}\right) \left[1 + \frac{\lambda_i - \lambda_1}{\lambda_1 - \omega}\right].
\end{aligned}$$

Increasing the distance $|\lambda_1 - \omega|$ between ω and λ_1 will minimise this ratio. To increase the gap between η_1 and the remainder of the spectrum of T_C we must therefore move ω away from λ_1.

Approach 2 It is convenient to write

$$\begin{aligned}
\eta_i &= \frac{\lambda_i - \theta}{\lambda_i - \omega} \\
&= \frac{\lambda_i - \omega + \omega - \theta}{\lambda_i - \omega} \\
&= 1 + \frac{\omega - \theta}{\lambda_i - \omega} \\
&= 1 - \frac{\theta - \omega}{\lambda_i - \omega}.
\end{aligned}$$

We assume that λ_i $(i = 2, \ldots, n)$ is not close to ω. Then we have $\eta_i \to 1$ as $\omega \to \theta$. Minimising the quantity $|\omega - \theta|$ thus maps the eigenvalues (except λ_1) of A to eigenvalues close to 1 of T_C.

2.2.5 The Inexact Cayley Transform

When the matrix $(A - \omega I)$ is inverted approximately in the Cayley transform we have an *inexact Cayley transform.* We use the notation $[A - \omega I]^{-1}$ to represent the action of $(A - \omega I)^{-1}$ when it is computed inexactly. For example, $x = [A - sI]^{-1}b$ might represent the approximate solution of the system $(A - sI)x = b$ using some Iterative Solver. With this notation we denote by M_C the inexact Cayley transform

$$M_C = [A - \omega I]^{-1}(A - \theta I).$$

Meerbergen [39, Ch.3] gives a detailed analysis of the spectral properties of the inexact Cayley transform, including the following theorem.

Theorem 2.2 (Meerbergen [39, Theorem 3.2.1])

For each simple eigenvalue η_i of T_C there is an eigenvalue μ_k of M_C such

$$|\eta_i - \mu_k| \leq |\eta_i| \mathrm{cond}_2(X) \|G\mathcal{P}_i\|$$

where $G = I - [A - \omega I]^{-1}(A - \omega I)$, X is the matrix of eigenvectors of M_C, and $\mathcal{P}_i$ is the spectral projector for η_i (see Section 1.1.1).

Theorem 2.2 indicates that if $\mathrm{cond}_2(X)$ is small then small eigenvalues of T_C will be perturbed by a small amount in the inexact Cayley transform. A corresponding result holds for the eigenvectors of M_C (Meerbergen [39, Theorem 3.2.2]).

2.2.6 Davidson's method

We return to our study of the spectrum of the matrix $(M - \theta I)^{-1}(A - \theta I)$ arising in the Generalised Davidson method. This generalises to the study of the matrix

$[A - \omega I]^{-1}(A - \theta I)$ which arises in both GD and Inverse Correction, and is an inexact Cayley transform.

We will temporarily assume, based on Theorem 2.2, that the eigenvalues of the inexact Cayley transform are close to those of the Cayley transform. Our analysis of the Cayley transform in Section 2.2.4 suggested two approaches:

Approach 1—ω away from λ_1 Here we attempt to maximise the separation of η_1 from the other eigenvalues of T_C. It is well established that techniques such as Subspace Iteration or Arnoldi's method converge more quickly to well separated eigenvalues. Since the Generalized Davidson method can be viewed asymptotically as an inexact Cayley Transform Arnoldi method [42] we expect that this shift selection approach to work well in GD.

Approach 2—ω close to λ_1 Here we attempt to place all except for one of the eigenvalues of T_C at 1. This does not maximise the separation of η_1 from the other eigenvalues of T_C. We now show that this is not necessary.

Let $\hat{x}$ be an approximate eigenvector and suppose that T_C has eigenvalue η_1 close to zero and its other eigenvalues $\eta_2, \ldots, \eta_n$ at 1. Writing $\hat{x}$ as a linear combination of the eigenvectors of A we have

$$\hat{x} = \sum_{i=1}^{n} \alpha_i x_i$$

for some scalars $\alpha_1, \ldots, \alpha_n$. Applying T_C to $\hat{x}$ gives

$$\begin{aligned} z &= T_C\hat{x} \\ &= T_C \sum_{i=1}^{n} \alpha_i x_i \\ &= \alpha_1 T_C x_1 + \sum_{i=2}^{n} \alpha_i T_C x_i \\ &= \alpha_1 \eta_1 x_1 + \sum_{i=2}^{n} \alpha_i \eta_i x_i \\ &= \alpha_1 \eta_1 x_1 + \sum_{i=2}^{n} \alpha_i x_i. \end{aligned}$$

It follows that

$$\begin{aligned} \hat{x} - z &= \hat{x} - T_C\hat{x} \\ &= \left(\alpha_1 x_1 + \sum_{i=2}^{n} \alpha_i x_i\right) - \left(\alpha_1 \eta_1 x_1 + \sum_{i=2}^{n} \alpha_i x_i\right) \\ &= (1 - \eta_1)\alpha_1 x_1. \end{aligned}$$

In this case we have computed the eigenvector x_1 in one step, even though the eigenvalue η_1 is not well separated. However, we make the following remark.

Remark

The eigenvalues at 1 of T_C are *not* close to the origin and so may be significantly perturbed in the inexact Cayley transform. To prevent this requires accurate solving of the linear system in the inexact Cayley transform (see Theorem 2.2).

Summary Approach 1 attempts to manipulate the spectrum of $[A - \omega I]^{-1}(A - \theta I)$ so that the eigenvalue η has separation suitable for application of Arnoldi's method or Subspace Iteration. In contrast, Approach 2 attempts to manipulate the spectrum of $[A - \omega I]^{-1}(A - \theta I)$ so that $[A - \omega I]^{-1}(A - \theta I)\hat{x}$ makes a good *correction.*

Approach 2, with σ close to 1, is particularly appropriate for Inverse Correction and for the Generalized Davidson method, when it is viewed as a correction method in

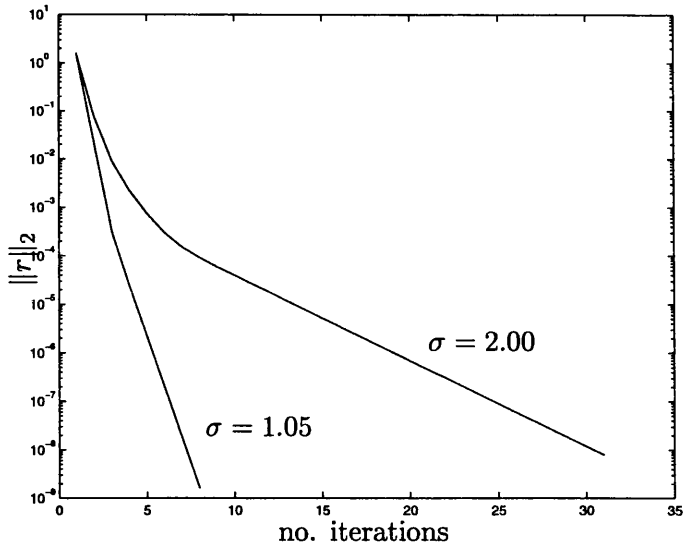


Figure 2-3: Residual norm plotted against number of iterations for Inverse Correction applied to the matrix in Example 2.1 with $\sigma = 1.05, 2.00$. Solves are performed with $\text{tol} = 1 \times 10^{-3}$.

which the correction is affected by the Rayleigh-Ritz procedure. Indeed, Morgan [42] and Morgan and Scott [43] choose $\omega = \theta$ in GD.

Example 2.2

Let A be the matrix A in Example 2.1. We now apply the Inverse Correction method with $\sigma = 2.00$ (Approach 1) and with $\sigma = 1.05$ (Approach 2). The residual norm is plotted against number of iterations in Figure 2-3. We see that the convergence of Inverse Correction is significantly slower with $\sigma = 2.00$ (Approach 1) than with $\sigma = 1.05$ (Approach 2).

2.3 The Jacobi-Davidson method

2.3.1 Davidson's method

Let $\hat{x}$ be an approximate eigenvector with Rayleigh Quotient θ and residual $r := A\hat{x} - \theta\hat{x}$. The correction vector computed by the Generalized Davidson method with

Algorithm 2.3: Jacobi-Davidson

Choose initial guess vector $\hat{x}$, and let $V = [\hat{x}]$.

1. Compute $\theta = \rho(\hat{x})$, and the residual $r = (A - \theta I)\hat{x}$.
2. For $k = 1, 2, \ldots$ do
   a) Solve
      $(I - \hat{x}\hat{x}^H)(A - \theta I)(I - \hat{x}\hat{x}^H)z = r$,
   b) Let $V = \text{mgs}([V, z])$,
   c) Compute the Ritz Pair $(\hat{x}, \theta)$ using the Rayleigh-Ritz procedure, and compute the residual $r = (A - \theta I)\hat{x}$,
   d) Test for convergence.

$\omega = \theta$ is

$$z = [A - \theta I]^{-1} r.$$

Typically z is computed by solving (inexactly)

$$(A - \theta I)z = r. \tag{2.3}$$

If this system is solved exactly then we obtain $z = \hat{x}$ and the method stagnates. By solving the system too accurately we impede convergence because z does not add enough new information to the trial space (see [46]).

To ensure that new information is added we can require that $z \perp \hat{x}$. Then $(I - \hat{x}\hat{x}^H)z = z$ and since $r \perp \hat{x}$ we also have $(I - \hat{x}\hat{x}^H)r = r$. Thus we can replace (2.3) by

$$(I - \hat{x}\hat{x}^H)(A - \theta I)(I - \hat{x}\hat{x}^H)z = r. \tag{2.4}$$

Replacing (2.3) by (2.4) in the Generalized Davidson method leads to the Jacobi-Davidson method (Olsen, Jørgensen, and Simons [46], Sleijpen and Van der Vorst [66], and Stathopoulos, Saad, and Fischer [69]). An implementation of the Jacobi-Davidson method is given in Algorithm 2.3.

2.3.2 Solving in Jacobi-Davidson

The solve in line (2a) of Algorithm 2.3 can be chosen from the following

(i) $(I - \hat{x}\hat{x}^H)(A - \theta I)(I - \hat{x}\hat{x}^H)z = r,$

(ii) $(A - \theta I)z = r - \epsilon\hat{x}$ where ϵ is computed to ensure $z \perp \hat{x}$.

(iii)

$$\begin{bmatrix} A - \theta I & \hat{x} \\ \hat{x}^H & 0 \end{bmatrix} \begin{bmatrix} z \\ \epsilon \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}.$$

It is easy to see that these systems are mathematically equivalent. More sophisticated formulations incorporating left eigenvectors can also be used (see Fokkema, Sleijpen, and Van der Vorst [24] and Sleijpen, Booten, Fokkema, and Van der Vorst [64]). We make the following remarks.

Remarks

1. Rearranging (ii), Meerbergen [39] shows that

$$z = [A - \theta I]^{-1}(A - (\theta + \epsilon)I)\hat{x}.$$

   The correction computed from (ii) is thus obtained by applying an inexact Cayley transform to $\hat{x}$.

2. Let y be the exact solution of $(A - \theta I)y = \hat{x}$. Now mgs($[V, z]$), with z computed by form (ii) using exact solves, is equivalent to mgs($[V, y]$). Thus Jacobi-Davidson is mathematically equivalent to the Accelerated Rayleigh Quotient Iteration [66].

3. A system equivalent to (iii) arises in computing the correction in Newton's method for the function $F : \mathbb{C}^{n+1} \to \mathbb{C}^{n+1}$ given by

$$F(x, \theta) = \begin{bmatrix} Ax - \theta x \\ -\frac{1}{2}x^H x + \frac{1}{2} \end{bmatrix} = 0.$$

   This relation allows proof of convergence of the Jacobi-Davidson method as an inexact Newton method in the case where solves are performed inexactly (see Sleijpen and Van der Vorst [65]).

2.3.3 Deflation for Jacobi-Davidson

We now suppose that Jacobi-Davidson has computed the m eigenvalues $\lambda_1, \ldots, \lambda_m$ of A, and their corresponding eigenvectors $x_1, \ldots, x_m$. Let $X = [x_1, \ldots, x_m]$. We may compute an incomplete QR-decomposition $X = ZU$ where Z is an $n \times m$ orthonormal matrix. The columns of Z are thus Schur vectors of A.

It is common, within the Power Method or Arnoldi's method, in such a situation to iterate using the *deflated* matrix $A(I - ZZ^H)$ (see Chapters 4 and 6, Saad [56]), or equivalently, $(I - ZZ^H)A(I - ZZ^H)$. This matrix has the eigenvalues $\lambda_{m+1}, \ldots, \lambda_n$ of A, and m zero eigenvalues whose eigenvectors are $x_1, \ldots, x_m$ and which arise because of the deflation.

Van der Vorst and Golub [18] and Fokkema et al. [24] show that Jacobi-Davidson can be applied to the projected matrix $(I - ZZ^H)(A - \theta I)(I - ZZ^H)$. The correction equation then becomes

$$(I - \hat{x}\hat{x}^H)(I - ZZ^H)(A - \theta I)(I - ZZ^H)(I - \hat{x}\hat{x}^H)z = r$$

which can be written in the form

$$(I - QQ^H)(A - \theta I)(I - QQ^H)\ z = r \tag{2.5}$$

where Q is an orthonormal matrix obtained from the incomplete QR-factorisation $[Z, \hat{x}] = QR$.

2.3.4 Jacobi-Davidson for the Generalized Eigenvalue problem

The Jacobi-Davidson method can also be applied to the generalized eigenvalue problem (GEVP)

$$Ax = \lambda Bx. \tag{2.6}$$

Recall that, if $\hat{x}$ is an approximate eigenvector of the matrix pencil (A, B) then the Rayleigh Quotient of $\hat{x}$ is $\rho(\hat{x}) = \hat{x}^H A\hat{x}/\hat{x}^H B\hat{x}$. The residual of the approximate

eigenpair (x, θ) is $r := Ax - \theta Bx$.

The Rayleigh-Ritz procedure for the GEVP

Recall that, given a subspace spanned by the columns of the orthonormal matrix V, the Rayleigh-Ritz procedure for the standard eigenvalue problem (say $B = I$) produces approximate eigenpairs of A by computing eigenpairs (y, θ) of the small matrix $V^H AV$. The pair (x, θ) with $x = Vy$ is an approximate eigenpair of A, and its residual is orthogonal to V.

In the same way, eigenpairs (y, θ) of the small generalized eigenvalue problem $V^H AVy = \theta\ V^H BV$ produce approximate eigenpairs for (2.6). If V is B-orthogonal, that is $V^H BV = I$, then the small GEVP reduces to the standard eigenvalue problem $V^H AVy = \theta y$.

Implementation of JD for the GEVP

An implementation of Jacobi-Davidson for the generalized eigenvalue problem is given in Algorithm 2.4. We observe that the system

$$(I - \hat{x}\hat{x}^H)(A - \theta I)(I - \hat{x}\hat{x}^H)z = r$$

in line (2a) of Algorithm 2.3 is replaced by the system

$$(I - wa^H)(A - \theta B)(I - bv^H)\ z = r \tag{2.7}$$

in Algorithm 2.4, where $v, w, a, b \in \mathbb{C}^n$ satisfy $a^H w = 1 = v^H b$. As for the standard eigenvalue problem, this system may be replaced by either

(i) the system $(A - \theta B)z = r - \epsilon w$ where ϵ is computed to ensure $z \perp v$.

(ii) the block system

$$\begin{bmatrix} A - \theta B & w \\ v^H & 0 \end{bmatrix} \begin{bmatrix} z \\ \epsilon \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}.$$

Algorithm 2.4: Jacobi-Davidson

Choose initial guess vector $\hat{x}$, and let $V = [\hat{x}]$.
Choose a, b, v, w—see equation (2.7).
1. Compute $\theta = \rho(\hat{x})$, and the residual $r = (A - \theta B)\hat{x}$.
2. For $k = 1, 2, \ldots$ do
   a) Solve
   $(I - wa^H)(A - \theta B)(I - bv^H)z = r$,
   b) Let $V = \text{mgs}([V, z])$,
   c) Compute the Ritz Pair $(\hat{x}, \theta)$ using the Rayleigh-Ritz procedure, and the residual $r = (A - \theta B)\hat{x}$,
   d) Test for convergence.

A number of choices for v and w (and a and b) are possible.

- Booten and Van der Vorst [6],[7] state that the choice $w = B\hat{x}$ gives quadratic convergence, with $a = b = \hat{x}$ and either $v = \hat{x}$ or $v = B^H\hat{x}$. This is a natural choice—the block system with $v = w = B\hat{x}$ arises in Newton's method for $Ax = \lambda Bx$.

- Booten et al. [8] present a JD algorithm for generalized eigenvalue problems in which B is Hermitian. This algorithm has $w = v = B\hat{x}$ and uses the B-orthogonal Rayleigh-Ritz procedure.

- Sleijpen et al. [64] present an analysis of JD for general v, w, a and b but recommend $v = B^H\hat{x}$, $w = B\hat{x}$ and $a = b = \hat{x}$.

2.3.5 Preconditioning Jacobi-Davidson

Booten and Van der Vorst [6, 7] recommend solving the systems of equations in Jacobi-Davidson inexactly, for example using GMRES. When the spectrum of $(A - \theta B)$ is poorly distributed (Chapter 3 gives a detailed discussion of how the spectrum of a matrix effects the convergence rate of GMRES) it may be necessary to precondition to improve the convergence rate of an iterative solver for (2.7).

Remarks

(i) If (2.7) is reformulated as $(A - \theta B)z = r - \epsilon w$ then preconditioning is easy since it is the matrix $(A - \theta B)$ which is to be preconditioned (see Sleijpen and Van

der Vorst [66]). However, we show in Chapter 3 that solves of the form (2.7) are better conditioned than solves with $(A - \theta B)$.

(ii) Booten and Van der Vorst [6, 7] precondition the block form

$$\begin{bmatrix} A - \theta B & w \\ v^H & 0 \end{bmatrix} \begin{bmatrix} z \\ \epsilon \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}.$$

Note that if inexact solves are used then z computed in this way is *not* orthogonal to $\hat{x}$.

The natural choice of preconditioner for (2.7) is of the form

$$(I - wa^H)M(I - bv^H) \tag{2.8}$$

where M, loosely speaking, would be a suitable preconditioner for $A - \theta B$. Sleijpen et al. [64, §7.1] show that the inverse of such a preconditioner, as a map from $b^\perp$ to $a^\perp$, is easy to evaluate—in fact the preconditioned form of (2.7) with such a preconditioner is

$$\left(I - \frac{\psi v^H}{v^H \psi}\right) M^{-1}(A - \theta B) \left(I - \frac{\psi v^H}{v^H \psi}\right) z = \left(I - \frac{\psi v^H}{v^H \psi}\right) M^{-1} r,$$

where $M\psi = w$ (see Theorem 7.3 of Sleijpen et al. [64]).

2.3.6 Preconditioning deflated Jacobi-Davidson

It is convenient to here remark that the correction equation (2.5)

$$(I - QQ^H)(A - \theta I)(I - QQ^H)\, z = r$$

in deflated Jacobi-Davidson may be preconditioned in the way described above (see Fokkema et al. [24]). If M is a preconditioner for $(A - \theta I)$ then the preconditioned

correction equation is

$$(I - YH^{-1}Q^H)M^{-1}(A - \theta I)(I - YH^{-1}Q)\ z = (I - YH^{-1}Q)M^{-1}r,$$

where $Y = M^{-1}Q$ and $H = Q^HY$.

2.4 The block eigenvalue problem

In the remainder of this chapter we consider Jacobi-Davidson for generalized eigenvalue problems which have a particular block structure. We show that with $w = B\hat{x}$ Jacobi-Davidson successfully computes approximate eigenpairs for these problems. In contrast, with $w = \hat{x}$ the Jacobi-Davidson iteration does not converge.

Let A and B be real $N \times N$ matrices with block structure

$$A = \begin{bmatrix} K & C \\ C^T & 0 \end{bmatrix}, \quad B = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}, \tag{2.9}$$

with A nonsingular and where K and positive definite M are real $n \times n$ matrices, and C is a real $n \times m$ full rank matrix. Generalized eigenvalue problems with this block structure arise in stability analysis for systems arising from mixed finite element discretizations of the Stokes and Navier-Stokes equations (see, for example, Meerbergen and Spence [40]).

The eigenvalue problem (2.6) has (finite) eigenvalues λ satisfying $\det(A - \lambda B) = 0$. Since B is singular (2.6) also has infinite eigenvalues with eigenvectors in $\mathcal{N}(B)$. A general theory for the generalized eigenvalue problem is given in Ericsson [22].

Meerbergen and Spence [40] rewrite (2.6) as

$$Tx = \mu x$$

where $T = A^{-1}B$. Finite eigenvalues λ of (2.6) correspond to eigenvalues $\mu = 1/\lambda$ of T. Infinite eigenvalues of (2.6) correspond to zero eigenvalues of T. Meerbergen and Spence give the following theorem on T.

Theorem 2.3 (Meerbergen and Spence [40, Theorem 1])

T has $n - m$ nonzero eigenvalues, and a zero eigenvalue of algebraic multiplicity $2m$ and geometric multiplicity m. The order of the Jordan blocks corresponding to the defective eigenvalue 0 is two.

Also, $\mathcal{N}(T) = \mathcal{N}(B)$ has dimension m, the generalized null space *$\mathcal{G} := \mathcal{N}(T^2) \setminus \mathcal{N}(T)$ has dimension m, and*

$$\mathbb{C}^N = \mathcal{R}(T) + \mathcal{N}(T) + \mathcal{G}.$$

Furthermore, $T\mathcal{G} = \mathcal{N}(T)$ and $T^2\mathcal{G} = T\,\mathcal{N}(T) = \{0\}$.

Remark

Meerbergen and Spence [40] point out that there is no loss in generality in considering T in place of $T_{SI} = (A - sB)^{-1}B$. For simplicity, we do so.

With A and B as given in (2.9) the eigenvalue problem (2.6) becomes

$$\begin{bmatrix} K & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \lambda \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix}. \tag{2.10}$$

Multiplying out we have

$$\begin{aligned} Ku + Cp &= \lambda Mu, \\ C^T u &= 0. \end{aligned}$$

The components u and p of x represent velocity and pressure components respectively. The velocity component satisfies the condition $C^T u = 0$ which is obtained from the incompressibility condition in the Stokes or Navier-Stokes equations.

2.4.1 Jacobi-Davidson for the block eigenvalue problem

When the subspace computed by JD contains approximations to null vectors of B (that is, eigenvectors corresponding to infinite eigenvalues of (A, B)) the Rayleigh-Ritz pro-

cedure computes *spurious* eigenvalues—approximations to the infinite eigenvalues of (2.6). These can be mistaken for approximations to finite eigenvalues. Discussions of how computation of spurious eigenvalues can be avoided for the shift-invert transformation can be found in Omid, Parlett, Ericsson and Jensen [45] (for the B-orthogonal Lanczos method) and Meerbergen and Spence [40] (for the B-orthogonal Implicitly Restarted Arnoldi method).

We now show that

(i) $\mathbb{C}^N$ can be decomposed into the null space of T, the generalized null space of T, and the space $\{x = (x_u^T, x_p^T)^T \in \mathbb{C}^N : C^T x_u = 0\}$.

(ii) if the initial guess vector $\hat{x}^{(0)}$ in JD with $w = B\hat{x}$ satisfies $C^T \hat{x}_u^{(0)} = 0$ then the subspace $V = [V_u^T, V_p^T]^T$ at a given step of JD satisfies $C^T V_u = 0$.

It follows that if $C^T \hat{x}_u^{(0)} = 0$ then the subspace computed by JD does not contain approximations to null vectors of B, and thus no spurious eigenvalues are computed.

Lemma 2.4

The generalized null space $\mathcal{N}(T^2) \setminus \mathcal{N}(T)$ *of* T *is* $\{x = (x_u^T, x_p^T)^T \in \mathbb{C}^N \setminus \mathcal{N}(T) : C^T x_u \neq 0\}$.

Proof We compute the null space of $T^2 = A^{-1}BA^{-1}B$. Since A is nonsingular $T^2 x = 0$ if and only if $BA^{-1}Bx = 0$.

We solve the system

$$\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} K & C \\ C^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_u \\ x_p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

by solving the systems

$$\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_u \\ y_p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{2.11}$$

$$\begin{bmatrix} K & C \\ C^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_u \\ x_p \end{bmatrix} = \begin{bmatrix} y_u \\ y_p \end{bmatrix}. \tag{2.12}$$

From (2.11) we have $y_u = 0$, but no constraint on y_p. Now (2.12) becomes

$$\begin{bmatrix} K & C \\ C^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_u \\ x_p \end{bmatrix} = \begin{bmatrix} 0 \\ y_p \end{bmatrix}.$$

so that

$$\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_u \\ x_p \end{bmatrix} = \begin{bmatrix} Cy_p \\ 0 \end{bmatrix}.$$

It follows that $Mx_u = Cy_p$. Thus x is in the null space of T^2 if and only if $x_u \in \{M^{-1}Cw \ : \ w \in \mathbb{C}^m\}$.

Let M have Choleski decomposition $M = LL^T$. Then

$$\begin{aligned} C^T M^{-1} C &= C^T L^{-T} L^{-1} C \\ &= (L^{-1}C)^T (L^{-1}C). \end{aligned}$$

It follows that $C^T M^{-1} C w = 0$ if and only if $w = 0$. □

We now show that, with exact solves, all vectors $x = (x_u^T, x_p^T)^T$ in the subspace computed by Jacobi-Davidson satisfy $C^T x_u = 0$.

Theorem 2.5 *Suppose that the initial guess vector $x = (x_u^T, x_p^T)^T$ satisfies $C^T x_u = 0$. Then the subspace $V_k = \left[V_u^T, V_p^T\right]^T$ at step k of Jacobi-Davidson (with $w = B\hat{x}$) satisfies $C^T V_u = 0$.*

Proof

Suppose that $V_k = \left[V_u^T, V_p^T\right]^T$ with $C^T V_u = 0$. Let $(\hat{x}, \theta)$ be a Ritz pair of (2.10). Then

$\hat{x} = Vy$ for some $y \in \mathbb{C}^k$ and so $C^T\hat{x}_u = C^T V_u y = 0$. The Ritz vector $\hat{x}$ has residual

$$r = \begin{bmatrix} r_u \\ r_p \end{bmatrix} = \begin{bmatrix} K & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_u \\ \hat{x}_p \end{bmatrix} - \theta \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_u \\ \hat{x}_p \end{bmatrix}$$
$$= \begin{bmatrix} K\hat{x}_u + C\hat{x}_p \\ 0 \end{bmatrix} - \theta \begin{bmatrix} M\hat{x}_u \\ 0 \end{bmatrix}.$$

Thus $r_p = 0$. The correction vector $z = (z_u^T, z_p^T)^T$ computed in Jacobi-Davidson satisfies

$$\begin{bmatrix} K - \theta M & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} z_u \\ z_p \end{bmatrix} = \begin{bmatrix} r_u \\ 0 \end{bmatrix} - \epsilon \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_u \\ \hat{x}_p \end{bmatrix}$$
$$= \begin{bmatrix} r_u \\ 0 \end{bmatrix} - \epsilon \begin{bmatrix} M\hat{x}_u \\ 0 \end{bmatrix}.$$

Thus $C^T z_u = 0$.

The result follows by induction. □

Remark

This result does not hold when $w = \hat{x}$ in Algorithm 2.4—it is easily seen that $r_p = \hat{x}_p$ so that $C^T z_u = z_p$.

2.4.2 Numerical Results

We now apply Jacobi-Davidson to an eigenvalue problem of the form (2.10) and compare the results for the choices $v = w = B\hat{x}$ and $v = w = \hat{x}$.

Example 2.3 Let $K = \text{diag}(1 : 50)$, $M = I_{50}$, and $C = [I_{10}, 0]^T$. The pencil (A, B) has finite eigenvalues $11, \ldots, 50$ and twenty infinite eigenvalues. Note that for this problem the null space of B is $\langle e_{51}, \ldots, e_{60} \rangle$.

We attempt to compute the rightmost finite eigenvalue using Jacobi-Davidson with (i) $v = w = B\hat{x}$ and (ii) $v = w = \hat{x}$. Results are plotted in Figure 2-4. We see that JD in case (i) computes an eigenpair in 25 steps, whilst JD in case (ii) fails. In fact JD

in case (ii) computes an infinite eigenvector—it's component in $\mathcal{N}(B)$ is indicated by $\|x_p\|_2$ which converges to 1. We also see that in case (i) the vector $C^T\hat{x}$ stays small, whilst in case (ii) this vector increases in magnitude. These results agree with the theory.

2.5 Summary

The stagnation of Inverse Iteration for some shifts when implemented using GMRES has been illustrated and explained. It is harder to prove results for other Krylov solvers that do not minimise the residual, but it is natural to expect stagnation for these also.

The Inverse Correction method does not stagnate when implemented using GMRES. By discussing the link between Inverse Correction and the Generalized Davidson method we have provided an alternative view of the shift selection strategy proposed by Rüde and Schmid. This strategy applies equally to the Generalized Davidson method, and contrasts with the traditional approach in iterative methods which attempts to maximise separation.

We have reviewed the myriad possible forms of the Jacobi-Davidson method, and considered their application to generalized eigenvalue problems of a special block structure that arise from the Stokes and Navier-Stokes equations. One particular form of the Jacobi-Davidson method is proven to prevent, in exact arithmetic, the computation of spurious eigenvalues. This is illustrated by a numerical example.

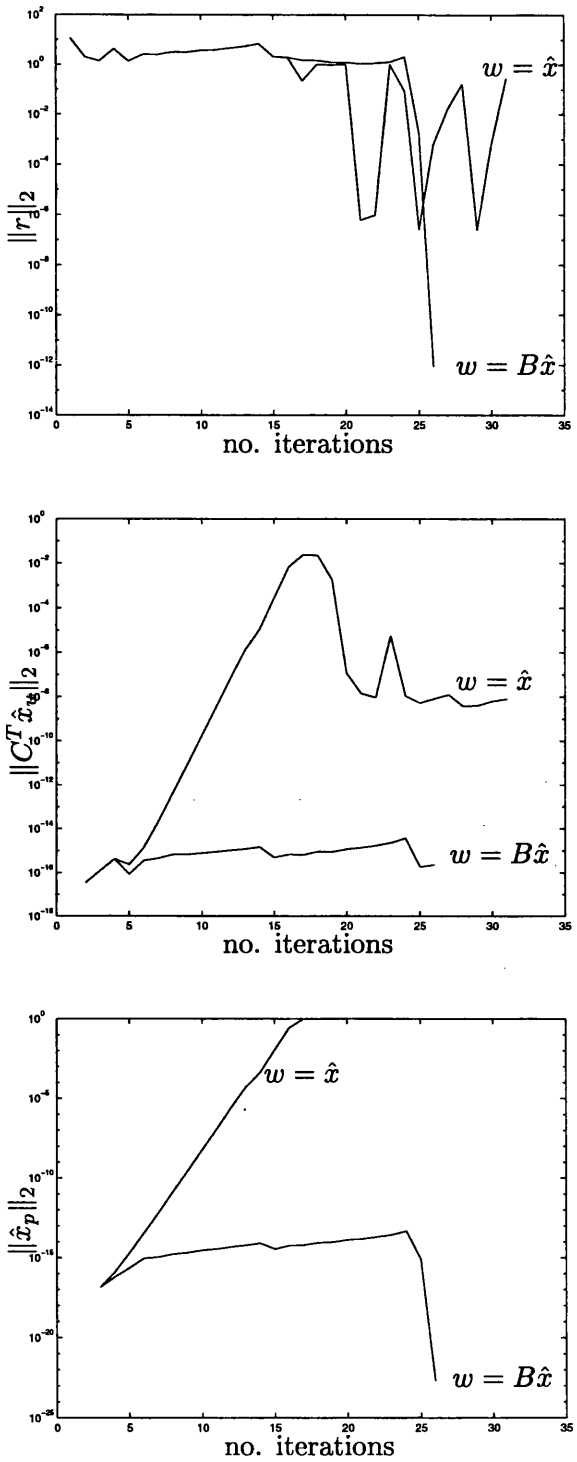


Figure 2-4: Residual norm (top), $\|C^T\hat{x}_u\|_2$ (middle), and $\|\hat{x}_p\|_2$ (bottom) for Example 2.3.

Chapter 3

GMRES for Projected Solves

3.1 Introduction

Is it possible to efficiently use iterative solvers to solve the near singular systems which arise when computing eigenvalues using methods based on the *Shift-Invert* transformation? We seek to reconcile two conflicting requirements: firstly we require that the shift be close to an eigenvalue for fast convergence of the eigenvalue solver; secondly, for fast convergence of the iterative solver we require that the shifted matrix have no small eigenvalues.

In this chapter we analyse the convergence rate of the GMRES algorithm, with particular attention to the dependence of the convergence rate on the eigenvalue distribution of the matrix. In Chapters 4 and 5 we will present methods based on the Shift-Invert transformation which satisfy the first of the above requirements, *and* efficiently use iterative solvers. With the analysis in this chapter we explain why the systems in Chapters 4 and 5 can be solved efficiently using GMRES [59]. This work also explains why the systems arising in the Jacobi-Davidson method [9], [24], [64], [66] can be solved efficiently using GMRES.

The outline of this chapter is as follows. In Section 3.2 we introduce the GMRES algorithm and using results due to Saad [59] and Chatelin [12] we discuss the relation between the eigenvalue distribution of the iteration matrix and the convergence rate of GMRES. In Section 3.3 we study the effects of small outlying eigenvalues on the

convergence rate of GMRES. Axelsson and Lindskog [2], Hackbusch [30], and Brooking [10] give results on the convergence rate of CG and GMRES with matrices which have outlying eigenvalues. We use these results to compare the convergence rate of GMRES for matrices with small outlying eigenvalues with the convergence rate of GMRES for matrices with the same spectrum, but no outlying eigenvalues. In Section 3.4 we show that a projection technique similar to Wielandt deflation (see Saad [56, Ch.2] or Wilkinson [75, Ch.9]) can be used to construct a system with the outlying eigenvalues effectively *removed.* We show in Theorem 3.15 that GMRES will compute a least squares solution for this projected system with a convergence rate independent of the outlying eigenvalues which are removed. The deflation is performed by means of projections which require knowledge of the exact eigenvectors corresponding to the outlying eigenvalues. In Section 3.4.5 we show that it is sufficient to use approximations to these eigenvectors, and Theorem 3.18, which is the main result of this chapter, tells us that GMRES will still compute a least squares solution for the projected system with a convergence rate independent of the eigenvalues which are removed. This analysis is illustrated by numerical examples.

3.2 GMRES convergence features

3.2.1 Krylov solvers

Let A be an $n \times n$, real or complex, large, square, *sparse* matrix. Given a complex vector b we seek the solution x of the linear system

$$Ax = b. \tag{3.1}$$

We do not consider *direct methods* here because they do not take advantage of the special sparse structure of A. Sparse structure reduces the cost of a matrix vector multiplication (mv) and makes *iterative methods* competitive.

Recall that for a complex vector v we define the *Krylov subspace* of dimension k,

generated by A and v, to be

$$\mathcal{K}_k(A, v) = \langle v, Av, \ldots, A^{k-1}v \rangle.$$

Krylov solvers compute an approximate solution for (3.1) by correcting an initial "guess vector". The correction vector is in a particular Krylov subspace—good Krylov subspaces lead to good approximate solutions which are close to the actual solution.

A measure of the quality of an approximate solution $\hat{x}$ is its *residual*

$$r = b - A\hat{x}.$$

The best measure of the quality of an approximate solution is its error $e = x - \hat{x}$. A small residual does not imply a small error, since $r = A^{-1}e$ and $\|A^{-1}\|_2$ may be large. However, residuals are easier to compute. Most Krylov solvers generate their Krylov subspace from the residual of the initial guess vector. The initial guess vector $x^{(0)}$ has initial residual $r^{(0)} = b - Ax^{(0)}$. We will use $z^{(k)}$ to denote the correction vector computed by the Krylov solver from the Krylov subspace $\mathcal{K}_k(A, r^{(0)})$. The corrected vector

$$x^{(k)} := x^{(0)} + z^{(k)}$$

is an approximate solution of (3.1). We say that $x^{(k)} \in x^{(0)} + \mathcal{K}_k(A, r^{(0)})$. Note here that a typical initial guess vector is zero. This has residual b.

In practice the Krylov subspace is successively extended by a new direction. This is an *iterative* process. We say that at *step* k the Krylov solver computes $x^{(k)}$ from the subspace $x^{(0)} + \mathcal{K}_k(A, r^{(0)})$. Examples of Krylov solvers are CG (Conjugate Gradients [32, 36]), FOM (Full Orthogonalisation Method), and GMRES (Generalised Minimum Residual). These Krylov solvers are described in Saad [58] and in Templates [5]. In this chapter we restrict our attention to GMRES [59].

3.2.2 GMRES

GMRES at step k computes an approximate solution $x^{(k)}$ of the system (3.1). The vector $x^{(k)}$ is the vector in $x^{(0)} + \mathcal{K}_k(A, r^{(0)})$ which has *smallest* residual (under the 2-norm). Thus $x^{(k)}$ satisfies

$$\|b - Ax^{(k)}\|_2 = \min_{x \in x^{(0)} + \mathcal{K}_k(A, r^{(0)})} \|b - Ax\|_2. \tag{3.2}$$

We may rewrite the residual of $x^{(k)}$ as

$$\begin{aligned} r^{(k)} &= b - Ax^{(k)} \\ &= b - A(x^{(0)} + z^{(k)}) \\ &= b - Ax^{(0)} - Az^{(k)} \\ &= r^{(0)} - Az^{(k)}. \end{aligned}$$

It is convenient to view each vector in $\mathcal{K}_k(A, r^{(0)})$ as the product of a particular polynomial of A with $r^{(0)}$. Let $\mathbb{P}_k$ denote the set of *polynomials* with complex coefficients and *degree* $\leq k$. Since $z^{(k)} \in \mathcal{K}_k(A, r^{(0)})$ there exists a polynomial $q \in \mathbb{P}_{k-1}$ such that $z^{(k)} = q(A)r^{(0)}$. With this notation

$$\begin{aligned} r^{(k)} &= r^{(0)} - Aq(A)r^{(0)} \\ &= (I - Aq(A))r^{(0)} \\ \text{and} \qquad \|r^{(k)}\|_2 &= \|(I - Aq(A))r^{(0)}\|_2. \end{aligned} \tag{3.3}$$

The polynomial $\tilde{p}(A) := (I - Aq(A))$ has degree $\leq k$, and satisfies $\tilde{p}(0) = 1$. From equation (3.2) and the last line of (3.3) we observe the important property of $\tilde{p}$ that

$$\|\tilde{p}(A)r^{(0)}\|_2 = \min_{\substack{p \in \mathbb{P}_k \\ p(0) = 1}} \|p(A)r^{(0)}\|_2.$$

As we shall see in Section 3.2.4, this property is fundamental to the convergence analysis of GMRES.

Algorithm 3.1: GMRES

Choose initial guess vector $x^{(0)}$,

1. Let $r^{(0)} = b - Ax^{(0)}$, $\beta = \|r^{(0)}\|_2$ and $v^{(1)} = r^{(0)}/\beta$.
2. For $j = 1, 2, \ldots, m$ do
   a) Let $w^{(j)} = Av^{(j)}$,
   b) For $i = 1, 2, \ldots, j$
      i) $h_{ij} = \langle v^{(i)}, w^{(j)} \rangle$,
      ii) $w^{(j)} = w^{(j)} - h_{ij} v^{(i)}$,
   c) Let $h_{j+1,j} = \|w^{(j)}\|_2$, $v^{(j+1)} = w^{(j)}/h_{j+1,j}$,
3. Compute $y^{(m)}$ which minimises $\|\beta e_1 - H_{m+1,m} y^{(m)}\|_2$ and $x^{(m)} = x^{(0)} + V_m y^{(m)}$.

3.2.3 Implementations of GMRES

We now briefly discuss the implementation of GMRES and present a GMRES algorithm for (3.1).

GMRES may be implemented with varying degrees of sophistication. Algorithm 3.1 is a simple implementation of GMRES which uses *modified Gram-Schmidt* (MGS) orthogonalisation. Saad [58, Algorithm 6.10] presents a more sophisticated implementation which uses *Householder* orthogonalisation. Householder orthogonalisation is numerically more robust than MGS.

Line two of Algorithm 3.1 implements *Arnoldi's method.* Arnoldi's method at the kth step computes an $n \times k$ orthonormal matrix V_k and a $k \times k$ *upper Hessenberg* matrix H_k. V_k and H_k have the property that

$$AV_k = V_k H_k + h_{k+1,k} v^{(k+1)} e_k^T. \tag{3.4}$$

This is called an *Arnoldi factorisation* of A. Equation (3.4) may alternatively be written

$$AV_k = V_{k+1} H_{k+1,k}$$

where $H_{k+1,k}$ is a $(k+1) \times k$ matrix obtained by adding the row $h_{k+1,k} e_k^T$ to H_k.

Line three of Algorithm 3.1 computes an approximate solution $x^{(m)}$ for (3.1). The approximate solution is given by $x^{(m)} = x^{(0)} + V_m y^{(m)}$ where $y^{(m)}$ is the *least squares*

solution of

$$H_{m+1,m}\, y^{(m)} = \beta e_1. \tag{3.5}$$

Left multiplication by V_{m+1} yields

$$AV_m y^{(m)} = r^{(0)}$$

and makes clear the link between the least squares solve (3.5) and the solve (3.1).

Algorithm 3.1 only computes a solution after m steps. In many practical implementations (3.5) is solved by transforming $H_{m+1,m}$ into *upper triangular* form using plane rotations. If the upper triangular form is stored then *only* one new plane rotation is required at each step. In this way (3.5) can be solved cheaply. In fact (3.5) can be solved cheaply enough that it is practical to compute an approximate solution at each step.

3.2.4 GMRES and the spectrum of A

Let λ be an *eigenvalue* of A, with corresponding *eigenvector* x. If p is a polynomial then $p(A)x = p(\lambda)x$. If p is small over the *spectrum* of A then $\|p(A)r^{(0)}\|_2$ will be small. This is qualified by Proposition 3.1.

We note here that if p is the minimum polynomial of A then $p(A)r^{(0)} = 0$ for any $r^{(0)}$. If the minimum polynomial of A has degree l then GMRES must compute at (or before) step l the exact solution of (3.1).

Proposition 3.1 (Saad [58, Proposition 6.15])

Assume that A is a diagonalisable matrix and let $A = X\Lambda X^{-1}$ where $\Lambda = \mathrm{diag}\{\lambda_1, \ldots, \lambda_n\}$ is the diagonal matrix of eigenvalues. Then the residual norm achieved at the kth step of GMRES satisfies the inequality

$$\|r^{(k)}\|_2 \leq \kappa_2(X) \min_{\substack{p \in \mathbb{P}_k \\ p(0)=1}} \max_{i=1,\ldots,n} |p(\lambda_i)| \|r^{(0)}\|_2.$$

Proof

We sketch the main details of this proof. The proof is given in full in Saad [58].

Let p be any polynomial of degree $\leq k$ satisfying $p(0) = 1$. Then

$$\begin{aligned} p(A)r^{(0)} &= p(X\Lambda X^{-1})r^{(0)} \\ &= Xp(\Lambda)X^{-1}r^{(0)} \\ \text{so} \quad \|p(A)r^{(0)}\|_2 &= \|Xp(\Lambda)X^{-1}r^{(0)}\|_2 \\ &\leq \|X\|_2 \, \|p(\Lambda)\|_2 \, \|X^{-1}\|_2 \, \|r^{(0)}\|_2 \\ &= \kappa_2(X) \, \|p(\Lambda)\|_2 \, \|r^{(0)}\|_2. \end{aligned}$$

Observe that $\|p(\Lambda)\|_2 = \max_{i=1,\ldots,n} |p(\lambda_i)|$. Hence

$$\|p(A)r^{(0)}\|_2 \leq \kappa_2(X) \max_{i=1,\ldots,n} |p(\lambda_i)| \|r^{(0)}\|_2.$$

This holds for any polynomial p of degree $\leq k$ satisfying $p(0) = 1$. Taking the minimum over all polynomials $p \in \mathbb{P}_k$ satisfying $p(0) = 1$ yields the result. □

Proposition 3.1 gives a bound which is, in some cases, pessimistic. Let $r^{(0)}$ have no component in, say the eigenvector x_j, of A. Then the magnitude of p at λ_j has no contribution to $\|r^{(k)}\|_2$ for any k. This is proved in Lemma 3.2. We will say that GMRES "*does not see*" the eigenvalue λ_j.

Lemma 3.2

Assume the conditions of Proposition 3.1. Without loss of generality, let $r^{(0)}$ have no component in the eigenvector x_n corresponding to the eigenvalue λ_n. Then the residual norm achieved at the kth step of GMRES satisfies the inequality

$$\|r^{(k)}\|_2 \leq \kappa_2(X) \min_{\substack{p \in \mathbb{P}_k \\ p(0)=1}} \max_{i=1,\ldots,n-1} |p(\lambda_i)| \|r^{(0)}\|_2.$$

Proof

This proof is similar to that of Proposition 3.1. Observe that we may write $X = [x_1, \ldots, x_n]$.

We notice that $XX^{-1}r^{(0)} = r^{(0)}$. Since $r^{(0)}$ has no component in x_n the vector $X^{-1}r^{(0)}$ has last entry zero. Thus $p(\Lambda)X^{-1}r^{(0)}$ has last entry zero.

The other entries of $p(\Lambda)X^{-1}r^{(0)}$ are independent of λ_n. It therefore follows that $\|Xp(\Lambda)X^{-1}r^{(0)}\|_2$ is independent of λ_n and hence

$$\|r^{(k)}\|_2 \leq \kappa_2(X) \min_{\substack{p \in \mathbb{P}_k \\ p(0)=1}} \max_{i=1,\ldots,n-1} |p(\lambda_i)| \|r^{(0)}\|_2.$$

□

3.2.5 Effects of eigenvalue distribution on the convergence of GMRES

In this section we discuss the convergence of GMRES for the system (3.1). Let b be a particular right hand side, and choose an initial guess vector $x^{(0)}$. The convergence behaviour of GMRES is now determined by the spectrum of A.

We present some examples illustrating the convergence behaviour of GMRES for some particular spectra. The conclusion of this section is a theorem which combines a number of well known results. These quantify the convergence rate of GMRES for (3.1) for particular eigenvalue distributions of A.

Recall the result of Proposition 3.1: that at step k of GMRES

$$\|r^{(k)}\|_2 \leq \kappa_2(X) \min_{\substack{p \in \mathbb{P}_k \\ p(0)=1}} \max_{i=1,\ldots,n} |p(\lambda_i)| \ \|r^{(0)}\|_2.$$

Let $D \subseteq \mathbb{C}$ be a domain which contains the spectrum of A. Then for $p \in \mathbb{P}_k$,

$$\max_{i=1,\ldots,n} |p(\lambda_i)| \leq \max_{z \in D} |p(z)|.$$

It follows that

$$\|r^{(k)}\|_2 \leq \kappa_2(X) \min_{\substack{p \in \mathbb{P}_k \\ p(0)=1}} \max_{z \in D} |p(z)| \|r^{(0)}\|_2. \tag{3.6}$$

We will refer to inequality (3.6) as the *minimax inequality.*

Definition 3.3

Let $p_k^\star$ denote the polynomial minimising $\max_{z \in D} |p(z)|$ over all polynomials $p \in \mathbb{P}_k$ satisfying $p(0) = 1$. We will call $p_k^\star$ the minimax polynomial.

Let $\tilde{p}_k$ denote the polynomial minimising $\|p(A)r^{(0)}\|_2$ over polynomials $p \in \mathbb{P}_k$ with $p(0) = 1$. By definition this is the polynomial computed by GMRES at step k. We will call $\tilde{p}_k$ the GMRES polynomial.

We hope that the bound

$$\max_{i=1,\ldots,n} |p(\lambda_i)| \leq \max_{z \in D} |p(z)|$$

will be sharp. If D only loosely encloses $\Lambda(A)$ then we cannot expect this. If D tightly encloses $\Lambda(A)$ then we can expect the bound to be sharp.

We examine the convergence of GMRES for each matrix A by examining the values of well chosen polynomials over domains which contain the spectrum of A. Sometimes we use the GMRES polynomials to illustrate the convergence behaviour. Sometimes the minimax polynomials are more appropriate.

Example 3.1

This example examines the convergence of GMRES when the eigenvalues of A are enclosed in a disk. We compare the convergence when this disk is close to the origin with the convergence when this disk is far from the origin. In this example the GMRES polynomials are complex. Complex polynomials are difficult to illustrate and we do not plot them. In Theorem 3.5 we will give the minimax polynomial for the disk, but we make use of it now. In the case of the disk the minimax polynomials are very simple.

1. Let the eigenvalues of A lie on the unit circle that is centred at 10. The degree k

minimax polynomial $p_k^\star$ for the domain $D(10,1)$ is $((z-10)/10)^k$. Its maximum value over $D(10,1)$ is $(1/10)^k$. The polynomial $p_3^\star$ is shown in Figure 3-1.

2. Let the eigenvalues of B lie on the unit circle that is centred at 2. The degree k minimax polynomial $q_k^\star$ over the domain $D(2,1)$ is $((z-2)/2)^k$. Its maximum value over $D(2,1)$ is $(1/2)^k$. The polynomial $q_3^\star$ is shown in Figure 3-1.

From the minimax inequality (3.6) we know that

$$\|r^{(k)}\|_2 \leq \kappa_2(X) \min_{\substack{p \in \mathbb{P}_k \\ p(0)=1}} \max_{z \in D} |p(z)| \, \|r^{(0)}\|_2.$$

If $\max_{z \in D} |p_k^\star(z)|$ is small then we expect $\|r^{(k)}\|_2$ to be small. If $\max_{z \in D} |p_k^\star(z)|$ is not small then we do not expect $\|r^{(k)}\|_2$ to be small.

In Figure 3-1 we see that, roughly speaking, $p_3^\star$ is small over $D(10,1)$. In comparison $q_3^\star$ is not small over $D(2,1)$. We expect that $\|r^{(k)}\|_2$ in case 1 is smaller than $\|r^{(k)}\|_2$ in case 2.

The convergence history for GMRES in these cases is shown in Figure 3-2. Note that, since in each case $\|r^{(k)}\|_2 = a^k$ for some constant a, we have that

$$\log_{10} \|r^{(k)}\|_2 = k \log_{10} a.$$

From the graph, the gradient $d(\log_{10} \|r^{(k)}\|_2)/dk$ for case 1 is -1. This yields $a = 1/10$. The gradient $d(\log_{10} \|r^{(k)}\|_2)/dk$ for case 2 is approximately $-.31$. This yields $a \approx 0.49$. The convergence bound given by the analysis closely matches that observed in practice.

Example 3.1 illustrates an important feature of the convergence of GMRES. If the spectrum of A lies away from the origin the convergence will typically be faster than if the spectrum lies close to the origin.

Note that GMRES computes different polynomials in case 1 and case 2. The matrices used were A and $B = A + 8I$ respectively. The Krylov subspaces computed in the two cases are identical, but the solutions are *different*—although the Krylov subspace

is *shift-invariant* GMRES is not.

Example 3.2

This example is very similar to Example 3.1. Let the spectrum of A lie in the real interval $[9, 11]$. Let the spectrum of B lie in the real interval $[1, 3]$. In this example the GMRES polynomials are real. The minimax polynomials are given in terms of Chebyshev polynomials which we discuss at the end of this section. We present the GMRES polynomials for this example instead of the minimax polynomials.

Let `A=diag(9:0.4:11)` and `B=A-8*I`. Let `b=ones(51,1)`. Figure 3-3 shows the GMRES polynomials $\tilde{p}_3$ and $\tilde{q}_3$ when GMRES is applied to $Ax = b$ and $Bx = b$ respectively. The GMRES polynomial $\tilde{p}_k$ is constructed from its roots which arise as eigenvalues of the generalised eigenvalue problem

$$H_{k+1,k}^H H_{k+1,k} \psi = \lambda H_k \psi.$$

This is discussed fully in Freund [27].

The convergence history for GMRES applied to $Ax = b$ and $Bx = b$ is shown in Figure 3-4. We see the same behaviour that we saw in Example 3.1—GMRES converges more quickly when the spectrum is further away from the origin.

For some domains minimax polynomials are known, for example, the disk $D(c, r)$. The degree k minimax polynomial over $D(c, r)$ is given by $((z-c)/r)^k$. This is proved in Chatelin [12]. Other common domains—in particular the ellipse and the real interval—have known minimax polynomials. To express these we first need the following definition.

Definition 3.4

The degree k Chebyshev polynomial $T_k(t)$ *is defined by*

$$T_k(t) = \begin{cases} \cos(k \cos^{-1} t), & \text{when } |t| \leq 1, \\ \cosh(k \cosh^{-1} t), & \text{when } |t| > 1. \end{cases}$$

The minimax polynomials for the ellipse and the real interval are given in terms

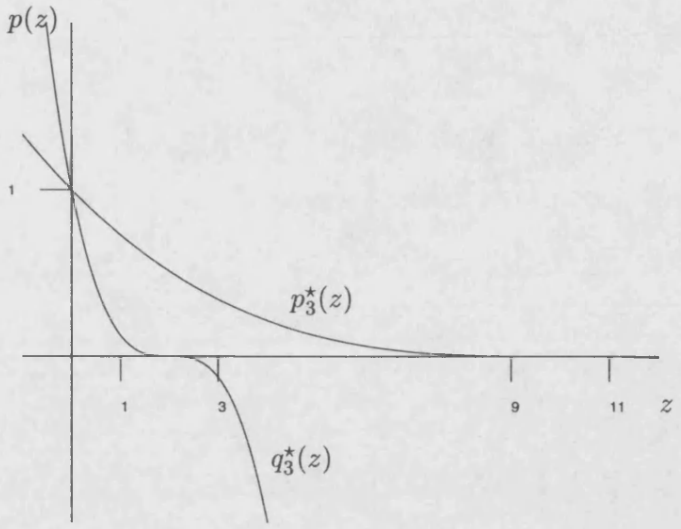


Figure 3-1: The minimax polynomials $p_3^\star(z)$ and $q_3^\star(z)$ for cases 1 and 2 of Example 3.1

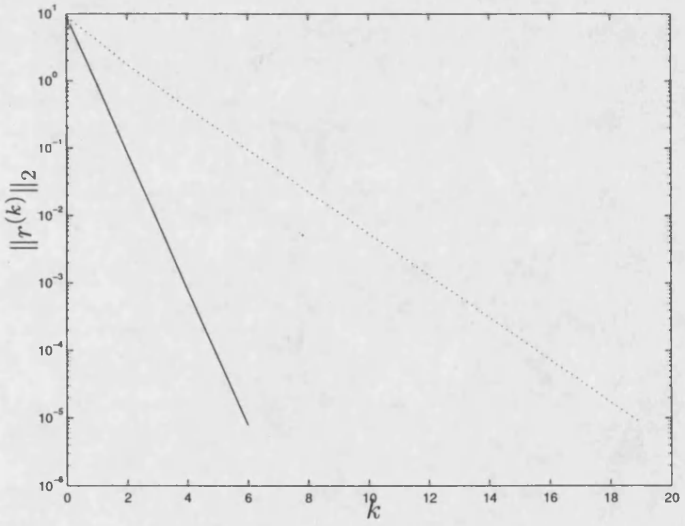


Figure 3-2: The residual norms $\|r^{(k)}\|_2$ from GMRES in case 1 (solid line) and case 2 (dashed line) of Example 3.1

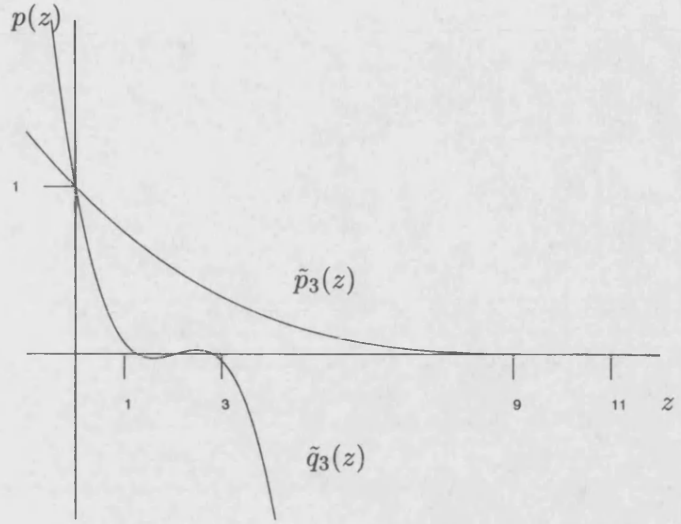


Figure 3-3: The GMRES polynomials $\tilde{p}_3(z)$ and $\tilde{q}_3(z)$ from Example 3.2

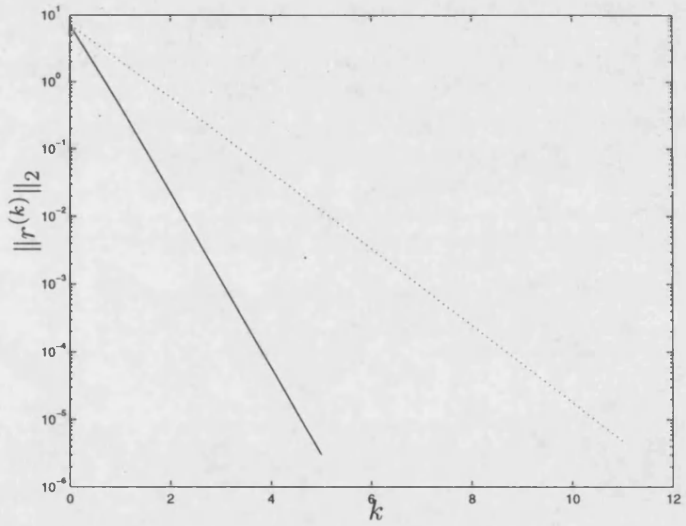


Figure 3-4: The residual norms $\|r^{(k)}\|_2$ from GMRES for $Ax = b$ (solid line) and $Bx = b$ (dashed line) of Example 3.2

of Chebyshev polynomials. Theorem 3.5 combines a number of results about minimax polynomials on these domains.

Theorem 3.5

Let D be a domain not containing the origin. Let r, c, d, and e be positive real numbers. Then

1. *when D is the real interval $[c-r, c+r]$ the degree k minimax polynomial for D is given by*

$$\frac{T_k((z-c)/r)}{T_k(c/r)}.$$

Its maximum value over D is $\frac{1}{T_k(c/r)}$.

2. *when D is the disk $D(c,r)$ the degree k minimax polynomial for D is given by*

$$\left(\frac{z-c}{c}\right)^k.$$

Its maximum value over D is $(r/c)^k$.

3. *when D is the ellipse $E(c,d,a)$ with centre c, focal distance d, and semi-major axis a, the degree k minimax polynomial for D is given by*

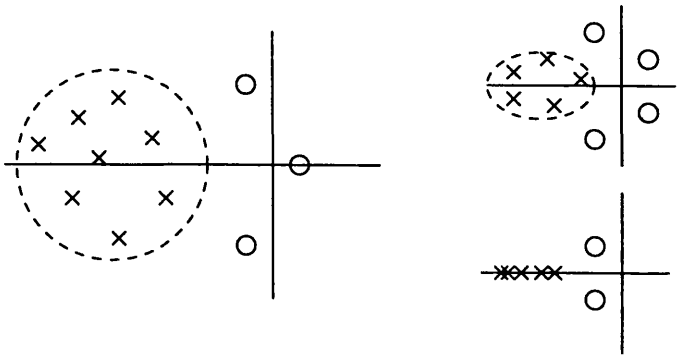
$$\frac{T_k((z-x)/d)}{T_k(c/d)}.$$

Its maximum value over D is $\frac{T_k(a/d)}{T_k(c/d)}$.

Proof

This theorem is a combination of Theorem 6.6.2, Theorem 7.2.1, Corollary 7.3.5, and Theorem 7.3.2 in Chatelin [12]. The construction of minimax polynomials is covered in detail in Chatelin, and also in Saad [58, Ch. 6]. □

When c is complex Theorem 3.5 cannot be applied; over complex domains the Chebyshev polynomials are *not* optimal. (This is discussed fully in Chatelin [12, Ch. 7]). However, it can be shown that the Chebyshev polynomials are asymptotically



× Enclosed eigenvalues

O Outlying eigenvalues

Figure 3-5: Some sample spectra with outlying eigenvalues.

optimal over complex domains. The bounds obtained from them are still useful, particularly when the degree of the polynomial is large.

3.3 Outlying eigenvalues

Suppose that most of the eigenvalues of the matrix A are grouped together. The remaining eigenvalues are the *outlying eigenvalues.* The outlying eigenvalues lie away from the rest of the spectrum. We will specifically use the term "outlying eigenvalues" for those eigenvalues which are not readily enclosed within a circle, or an ellipse (or a real interval) which contains most of the eigenvalues of A. Some spectra with outlying eigenvalues are illustrated in Figure 3-5.

When a matrix A has no outlying eigenvalues it is easy to estimate the convergence rate of GMRES for (3.1) using Theorem 3.5, Proposition 3.1, and the minimax inequality (3.6). (It is *always* possible to enclose the spectrum of A within a circle (or an ellipse). It is then usually [1] possible to bound the residual $\|r^{(k)}\|_2$ using the minimax inequality (3.6)). If A has outlying eigenvalues then it is not possible to *tightly* enclose the spectrum of A within a circle (or an ellipse). Consequently the residual bound obtained from the minimax inequality will not be sharp.

Let $S^{\mathcal{O}}$ be the set $\{\xi_1, \xi_2, \ldots, \xi_l\}$ of outlying eigenvalues of A. Let S be a domain

[1] If the spectrum is enclosed by a domain containing the origin then Theorem 3.5 does not apply.

which contains the rest of the spectrum of A. S corresponds to the regions enclosed by the dashed lines in Figure 3-5.

Brooking [10, Ch. 2] analyses the convergence of GMRES when A has outlying eigenvalues. He does this by finding polynomials which are small over $D := S \cup S^{\mathcal{O}}$. Note that $\Lambda(A) \subseteq D$ and so the minimax inequality (3.6) holds. A similar technique is used by Hackbusch [30, Ch. 9] to study convergence rates of CG for indefinite matrices that have almost all of their eigenvalues in one half plane, with a small number of eigenvalues in the other half plane. This technique is also used by Axelsson and Lindskog [2] to analyse the convergence of CG for matrices with outlying eigenvalues.

Axelsson, Hackbusch, and Brooking construct polynomials which are zero on $S^{\mathcal{O}}$, and small on S. Such polynomials may be constructed as the product of the linear factors $\left(1 - t/\xi_i\right), \quad i = 1, \ldots, l$, and the minimax polynomial for S described in Theorem 3.5.

We introduce some notation to describe these polynomials. Let $\widehat{\mathbb{P}}_k^l$ denote the set of polynomials p in $\mathbb{P}_{k+l}$ which satisfy $p(0) = 1$, and $p(\xi_i) = 0, \quad i = 1, \ldots, l$.

Let $p_k^\star$ denote the degree k minimax polynomial for the domain S. Note that $p_k^\star \in \mathbb{P}_k$, and that $p_k^\star \notin \widehat{\mathbb{P}}_k^l$. For the remainder of this section we will slightly abuse our previous notation; let $q_{k+l}^\star$ denote the polynomial which is the minimiser over $q \in \widehat{\mathbb{P}}_k^l$ of

$$\max_{z \in S} |q(z)|.$$

We will call $q_{k+l}^\star$ the *pseudo-minimax* polynomial for D. It is not the minimax polynomial for D; that polynomial is not readily computable. However, we expect that the pseudo-minimax polynomial is close to the minimax polynomial.

Observe that

$$\begin{aligned}
\max_{z \in D} |q_{k+l}^\star(z)| &= \max \left\{ \max_{z \in S^{\mathcal{O}}} |q_{k+l}^\star(z)|, \max_{z \in S} |q_{k+l}^\star(z)| \right\} \\
&= \max \left\{ 0, \max_{z \in S} |q_{k+l}^\star(z)| \right\} \\
&= \max_{z \in S} |q_{k+l}^\star(z)|.
\end{aligned}$$

In practice it is difficult to compute $\max_{z\in D}|q^\star_{k+l}(z)|$ since the maxima and minima of $q^\star_k$ are not maxima and minima of $p^\star_k$. Theorem 3.6 allows us to estimate $\max_{z\in D}|q^\star_{k+l}(z)|$.

Theorem 3.6 (Brooking [10])

Let $D = S^{\mathcal{O}} \cup S$, and let $p^\star_k \in \mathbb{P}_k$ be the degree k minimax polynomial for S. Let $q^\star_{k+l} \in \widehat{\mathbb{P}}^l_k$ be the degree $k+l$ pseudo-minimax polynomial for D. Then

$$\max_{z\in S}|p^\star_k(z)|\left[\prod_{i=1}^{l}\min_{z\in S}|1-z/\xi_i|\right] \le \max_{z\in D}|q^\star_{k+l}(z)| \le \max_{z\in S}|p^\star_k(z)|\left[\prod_{i=1}^{l}\max_{z\in S}|1-z/\xi_i|\right]. \tag{3.7}$$

Proof

Let $q \in \widehat{\mathbb{P}}^l_k$. Then we may write $q = f\cdot g$ where $f\in\mathbb{P}_k$ and $g = \prod_{i=1}^{l}(1+z/\xi_i)$. Since $g(0)=1$ and $q(0)=1$ we note that $f(0)=1$. Then (see also Brooking [10, Lemma 2.4])

$$\begin{aligned}
\min_{q\in\widehat{\mathbb{P}}^l_k}\max_{z\in S}|q(z)| &= \min_{\substack{f\in\mathbb{P}_k\\ f(0)=1}}\max_{z\in S}\left|f(z)\prod_{i=1}^{l}(1-z/\xi_i)\right| \\
&\le \min_{\substack{f\in\mathbb{P}_k\\ f(0)=1}}\left(\max_{z\in S}|f(z)|\right)\left(\max_{z\in S}\left|\prod_{i=1}^{l}(1-z/\xi_i)\right|\right) \\
&\le \left(\max_{z\in S}\left|\prod_{i=1}^{l}(1-z/\xi_i)\right|\right)\left(\min_{\substack{f\in\mathbb{P}_k\\ f(0)=1}}\max_{z\in S}|f(z)|\right) \\
&\le \left(\max_{z\in S}\left|\prod_{i=1}^{l}(1-z/\xi_i)\right|\right)\max_{z\in S}|p^\star_k(z)| \\
&\le \left(\prod_{i=1}^{l}\max_{z\in S}|1-z/\xi_i|\right)\max_{z\in S}|p^\star_k(z)|.
\end{aligned}$$

Similarly

$$\begin{aligned}
\min_{q\in\widehat{\mathbb{P}}_k^l}\max_{z\in S}|q(z)| &= \min_{\substack{f\in\mathbb{P}_k\\ f(0)=1}}\max_{z\in S}\left|f(z)\prod_{i=1}^{l}(1-z/\xi_i)\right| \\
&\geq \min_{\substack{f\in\mathbb{P}_k\\ f(0)=1}}\left(\max_{z\in S}|f(z)|\right)\left(\min_{z\in S}\left|\prod_{i=1}^{l}(1-z/\xi_i)\right|\right) \\
&\geq \left(\min_{z\in S}\left|\prod_{i=1}^{l}(1-z/\xi_i)\right|\right)\left(\min_{\substack{f\in\mathbb{P}_k\\ f(0)=1}}\max_{z\in S}|f(z)|\right) \\
&\geq \left(\min_{z\in S}\left|\prod_{i=1}^{l}(1-z/\xi_i)\right|\right)\max_{z\in S}|p_k^\star(z)| \\
&\geq \left(\prod_{i=1}^{l}\min_{z\in S}|1-z/\xi_i|\right)\max_{z\in S}|p_k^\star(z)|.
\end{aligned}$$

By Definition, $\max_{z\in D}|q_{k+l}^\star(z)| = \min_{q\in\widehat{\mathbb{P}}_k^l}\max_{z\in D}|q(z)|$. Finally we notice that, for polynomials $q\in\widehat{\mathbb{P}}_k^l$, $\max_{z\in D}|q(z)| = \max_{z\in S}|q(z)|$. □

3.3.1 Outlying eigenvalues - some examples

We complete this section on outlying eigenvalues by considering some applications of Theorem 3.6. In these applications we consider the convergence of GMRES when A has outlying eigenvalues. We compare this convergence with the convergence of GMRES for solves where the outlying eigenvalues are removed.

For each application we give an example. In the choice of examples we have in mind the applications in Chapters 4 and 5 where we present eigenvalue solvers which remove outlying eigenvalues with a view to increasing the rate of convergence of GMRES. GMRES is used in these eigenvalue solvers as an inner iteration.

For convenience we introduce some new notation. Suppose that A has outlying eigenvalues $\xi_1,\ldots,\xi_l$, and that $S^{\mathcal{O}} = \{\xi_1,\ldots,\xi_l\}$. Let S be a domain which contains the rest of the spectrum of A and let $D = S\cup S^{\mathcal{O}}$.

We define the *intermediate domains* S_j by $S_j = S\cup\{\xi_{j+1},\ldots,\xi_l\}$ for $j = 1,\ldots,l-1$.

It is also helpful to define $S_0 = D$ and $S_l = S$. Then

$$S = S_l \subseteq S_{l-1} \subseteq \cdots \subseteq S_1 \subseteq S_0 = D.$$

For each intermediate domain we introduce the *intermediate matrix* A_j. The matrix A_j has eigenvalues $\Lambda(A) \setminus \{\xi_1, \ldots, \xi_j\}$, that is, that A_j has j eigenvalues removed. Note that for each j we have $\Lambda(A_j) \subseteq S_j$. We do not discuss here the technicalities of creating the intermediate matrices; they are not unique and there are a number of possible constructions. We do not assume that the intermediate matrices are all of the same size.

Remark

The convergence rate of GMRES is independent of n, the size of the matrix. To see this, observe that the convergence rate of GMRES for a given right hand side is determined by the distribution of the eigenvalues of A (see Proposition 3.1).

Each intermediate domain S_j has a pseudo-minimax polynomial. It is convenient to denote by $q_k^{(j)}$ the degree k pseudo-minimax polynomial for S_j. This is consistent with the notation used for the intermediate domains. We will sometimes in examples with l outlying eigenvalues denote the minimax polynomial $p_k^\star$ for S_l by $q_k^{(l)}$.

We will use the pseudo-minimax polynomials for the intermediate domains S_j to estimate and compare the convergence rates of GMRES for the matrices A_j.

Notice from Theorem 3.6 that $\max_{z \in S} |q_{k+l}^\star(z)|$ is bounded above and below by fixed multiples of $\max_{z \in S} |p_k^\star(z)|$. Thus GMRES for the matrix A_0 with l outlying eigenvalues converges like GMRES for the matrix A_l which has the l outlying eigenvalues removed. But, *it is l steps behind.*

Application 1. S is a real interval We first consider a simple example. Let A_0 have the two real outlying eigenvalues ξ_1 and ξ_2 and let the other eigenvalues lie in the real interval $[\alpha, \beta]$. This is illustrated in Figure 3-6. We restrict ourselves to the case where $\alpha < \beta < 0$. This generalises to the case where α and β have the *same* sign.

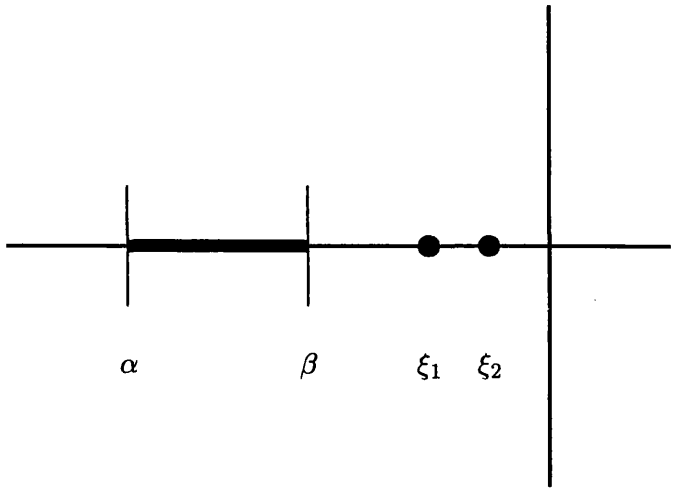


Figure 3-6: The matrix A has most of its spectrum in the interval $[\alpha, \beta]$. A also has the two outlying eigenvalues ξ_1 and ξ_2.

Using our previous notation we have

$$\begin{aligned} S_2 &= S = [\alpha, \beta] \\ S_1 &= S \cup \{\xi_2\} \\ S_0 &= S \cup \{\xi_2, \xi_1\}. \end{aligned}$$

It is convenient to notice that S has centre $c = (\alpha + \beta)/2$ and radius $r = (\beta - \alpha)/2$. By Theorem 3.5 the minimax polynomial of degree k for S is given by

$$p_k^\star(z) = \frac{T_k((z-c)/r)}{T_k(c/r)}.$$

This polynomial attains the maximum value $\frac{1}{T_k(c/r)}$ over S.

The pseudo-minimax polynomial $q_{k+2}^{(0)}(z)$ for the domain S_0 which has no eigenvalues removed (and hence two outlying eigenvalues) is given by

$$(1 - z/\xi_1)(1 - z/\xi_2)p_k^\star(z).$$

The pseudo-minimax polynomial $q_k^{(1)}(z)$ for the domain S_1 which has one eigenvalue

removed (and hence one outlying eigenvalue) is given by

$$(1 - z/\xi_2)p_k^\star(z).$$

It is easy to see that

$$\begin{aligned}
\min_{z\in S}|1 - z/\xi_2| &= |1 - \beta/\xi_2| \\
&= \left|\frac{1}{\xi_2}\right|(\xi_2 - \beta) \\
\max_{z\in S}|1 - z/\xi_2| &= |1 - \alpha/\xi_2| \\
&= \left|\frac{1}{\xi_2}\right|(\xi_2 - \alpha) \\
\min_{z\in S}|(1 - z/\xi_2)(1 - z/\xi_1)| &= (1 - \beta/\xi_2)(1 - \beta/\xi_1) \\
&= \left|\frac{1}{\xi_1\xi_2}\right|(\xi_2 - \beta)(\xi_1 - \beta) \\
\max_{z\in S}|(1 - z/\xi_2)(1 - z/\xi_1)| &= (1 - \alpha/\xi_2)(1 - \alpha/\xi_1) \\
&= \left|\frac{1}{\xi_1\xi_2}\right|(\xi_2 - \alpha)(\xi_1 - \alpha).
\end{aligned}$$

Theorem 3.6 gives the following bounds on the maxima of the pseudo-minimax polynomials.

$$\left|\frac{1}{\xi_2}\right||\xi_2 - \beta| \cdot \max_{z\in S}|p_{k-1}^\star(z)| \le \max_{z\in S}|q_k^{(1)}(z)| \le \left|\frac{1}{\xi_2}\right||\xi_2 - \alpha| \cdot \max_{z\in S}|p_{k-1}^\star(z)| \tag{3.8}$$

and with $K = |1/\xi_1\xi_2|$,

$$K(\xi_2 - \beta)(\xi_1 - \beta)\max_{z\in S}|p_{k-2}^\star(z)| \le \max_{z\in S}|q_k^{(0)}(z)| \le K(\xi_2 - \alpha)(\xi_1 - \alpha)\max_{z\in S}|p_{k-2}^\star(z)|. \tag{3.9}$$

It is clear that GMRES with the matrix A_0 lags j steps behind GMRES with the matrix A_j. In addition, if ξ_1 and ξ_2 are small then $1/\xi_2$ and $1/\xi_1\xi_2$ are large. Thus the residual norm at step k of GMRES for the matrix A_0 is a large multiple of the residual norm at step $k - j$ of GMRES for the matrix A_j.

A particular example of the case when A has two small outlying eigenvalues is given

in Example 3.3

Example 3.3

In this example we compare GMRES convergence rates for problems with zero, one, and two outlying eigenvalues. The solutions computed in each case are different but this is of no concern since we have a particular application in mind where this is not a problem.

Let `d` be the vector generated by `d=-(3:0.05:4)`. Let

```
A0 = diag([d,-0.05,-0.1])        b0 = ones(23,1)
A1 = diag([d,-0.1])              b1 = ones(22,1)
A2 = diag(d)                     b2 = ones(21,1).
```

The matrix `A0` has 2 outlying eigenvalues. The matrix `A1` has the spectrum of `A0` but with one outlying eigenvalue removed. The matrix `A2` has the spectrum of `A0` but with 2 eigenvalues removed.

We apply GMRES to the solves

```
A0 x0 = b0
A1 x1 = b1
A2 x2 = b2.
```

With the previous notation we have $\xi_1 = -0.05$, $\xi_2 = -0.1$, $\alpha = -3$ and $\beta = -4$. It follows that

$$\begin{aligned}
\left|\frac{1}{\xi_1\xi_2}\right|(\xi_2-\beta)(\xi_1-\beta) &= 200\cdot 2.9\cdot 2.95 \\
&= 1711 \\
\left|\frac{1}{\xi_1\xi_2}\right|(\xi_2-\alpha)(\xi_1-\alpha) &= 200\cdot 3.9\cdot 3.95 \\
&= 3081 \\
\left|\frac{1}{\xi_2}\right|(\xi_2-\beta) &= 10\cdot 2.9 \\
&= 29 \\
\left|\frac{1}{\xi_2}\right|(\xi_2-\alpha) &= 10\cdot 3.9 \\
&= 39.
\end{aligned}$$

From (3.9) we obtain the bound

$$29 \cdot \max_{z\in[-4,-3]} |q_{k-1}^{(2)}(z)| \leq \max_{z\in[-4,-3]} |q_k^{(1)}(z)| \leq 39 \cdot \max_{z\in[-4,-3]} |q_{k-1}^{(2)}(z)| \tag{3.10}$$

and from (3.8) we obtain the bound

$$1711 \cdot \max_{z\in[-4,-3]} |q_{k-2}^{(2)}(z)| \leq \max_{z\in[-4,-3]} |q_k^{(0)}(z)| \leq 3081 \cdot \max_{z\in[-4,-3]} |q_{k-2}^{(2)}(z)|. \tag{3.11}$$

Notice that A is *normal*. Thus the matrix of eigenvectors in the minimax inequality (3.6) has condition number $\kappa_2(X) = 1$. Let $r_0^{(k)}, r_1^{(k)}$ and $r_2^{(k)}$ denote the residuals of the approximations $x_0^{(k)}, x_1^{(k)}$ and $x_2^{(k)}$ to `x0`, `x1` and `x2` respectively at step k. If the pseudo-minimax inequality gives a tight bound here then by (3.10) we expect

$$29 \cdot \|r_2^{(k-1)}\|_2 \leq \|r_1^{(k)}\|_2 \leq 39 \cdot \|r_2^{(k-1)}\|_2 \tag{3.12}$$

and by (3.11) we expect

$$1711 \cdot \|r_2^{(k-2)}\|_2 \leq \|r_0^{(k)}\|_2 \leq 3081 \cdot \|r_2^{(k-2)}\|_2. \tag{3.13}$$

With each intermediate matrix there is a corresponding intermediate domain. For a small residual the GMRES polynomial for each intermediate matrix must be small over its corresponding intermediate domain.

Figures 3-7, 3-8, and 3-9 show the GMRES polynomials for `A0`, `A1`, and `A2` at steps $k = 1, 2, 3$ and 4.

Recall that the degree $k+l$ GMRES polynomial lies in $\widehat{\mathbb{P}}_k^l$ when it is zero at each of the outlying eigenvalues $\xi_1, \ldots, \xi_l$. Eventually the degree $k+l$ GMRES polynomial for the intermediate matrix `A1` will lie in $\widehat{\mathbb{P}}_k^l$. For a small residual the GMRES polynomial need then be small only on $[-4, -3]$.

Consider the solve `A2 x2 = b2`. The intermediate domain for `A2` is $S_2 = [-4, -3]$. `A2` has *no* outlying eigenvalues. Figure 3-7 shows some of the GMRES polynomials $\tilde{p}^{(2)}$ for `A2 x2 = b2`. The polynomials $\tilde{p}^{(2)}$ are small over S_2.

Consider the solve `A1 x1 = b1`. The intermediate domain for `A1` is $S_1 = [-4, -3] \cup$

$\{-0.1\}$. A1 has *one* outlying eigenvalue. Figure 3-8 shows some of the GMRES polynomials $\tilde{p}^{(1)}$ for `A1 x1 = b1`. The polynomials $\tilde{p}^{(1)}$ are of moderate size over $[-4,-3]$. At step $k=3$, $\tilde{p}_3^{(1)}$ has a root at -0.1. In fact $\tilde{p}_k^{(1)} \in \widehat{\mathbb{P}}_{k-1}^1$ for $k \geq 3$.

Consider the solve `A0 x0 = b0`. The intermediate domain for A0 is $S_0 = [-4,-3] \cup \{-0.1,-0.05\}$. A0 has *two* outlying eigenvalues. Figures 3-9 and 3-10 show some of the GMRES polynomials $\tilde{p}^{(0)}$ for `A0 x0 = b0`. The polynomials $\tilde{p}^{(0)}$ are not small over $[-4,-3]$ until after step $k=7$. At step $k=6$, $\tilde{p}_3^{(0)}$ has a root at -0.1 and a root at -0.05. In fact $\tilde{p}_k^{(0)} \in \widehat{\mathbb{P}}_{k-2}^1$ for $k \geq 6$.

Figure 3-11 shows the convergence history of GMRES with the matrices A2, A1 and A0. By (3.12) we expect $\|r_1^{(k)}\|_2 \approx c_1 \|r_2^{(k-1)}\|_2$ for $k \geq 3$, where $29 \leq c_1 \leq 39$. Thus

$$\begin{aligned} \log_{10} \|r_1^{(k)}\|_2 &\approx \log_{10} c_1 \|r_2^{(k-1)}\|_2 \\ &\approx \log_{10} c_1 + \log_{10} \|r_2^{(k-1)}\|_2. \end{aligned}$$

Then $\log_{10} \|r_1^{(k)}\|_2 \approx 1.5 + \log_{10} \|r_2^{(k-1)}\|_2$. A similar argument for the case with no outlying eigenvalues removed yields $\log_{10} \|r_0^{(k)}\|_2 \approx 3.3 + \log_{10} \|r_2^{(k-2)}\|_2$. In Figure 3-12 the values of $\log_{10} \|r_2^{(5)}\|_2$, $\log_{10} \|r_1^{(6)}\|_2$, and $\log_{10} \|r_0^{(7)}\|_2$ are marked. It is clear that the numerical results fit these theoretical estimates.

Table 3.1 displays the ratios

$$\rho_0^{(k)} := \frac{\|r_0^{(k+2)}\|_2}{\|r_2^{(k)}\|_2} \quad \text{and} \quad \rho_1^{(k)} := \frac{\|r_1^{(k+1)}\|_2}{\|r_2^{(k)}\|_2}.$$

We see that

$$\begin{aligned} \lim \rho_0^{(k)} &\approx 2270 \\ \lim \rho_1^{(k)} &\approx 33.5. \end{aligned}$$

These values fall within the bounds given by (3.12) and (3.13).

In Application 2 we consider problems where the spectrum of the matrix A lies in a disk centred on the real axis. A also has 2 complex conjugate outlying eigenvalues.

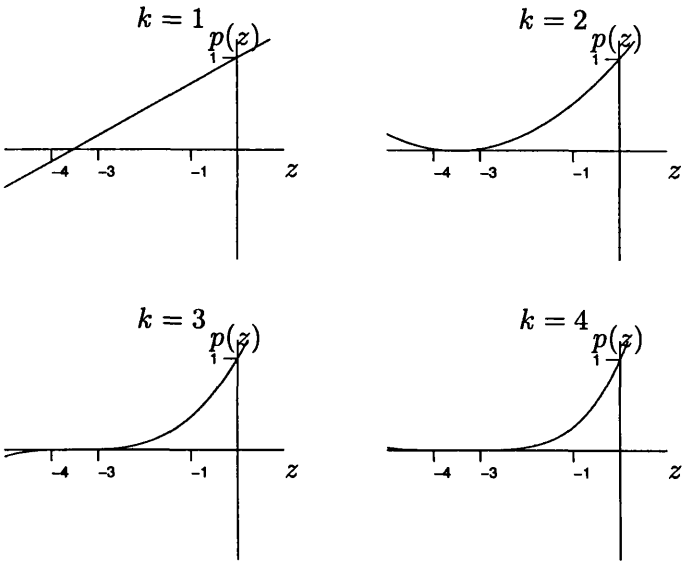


Figure 3-7: The GMRES polynomials for `A2 x2 = b2` from Example 3.3

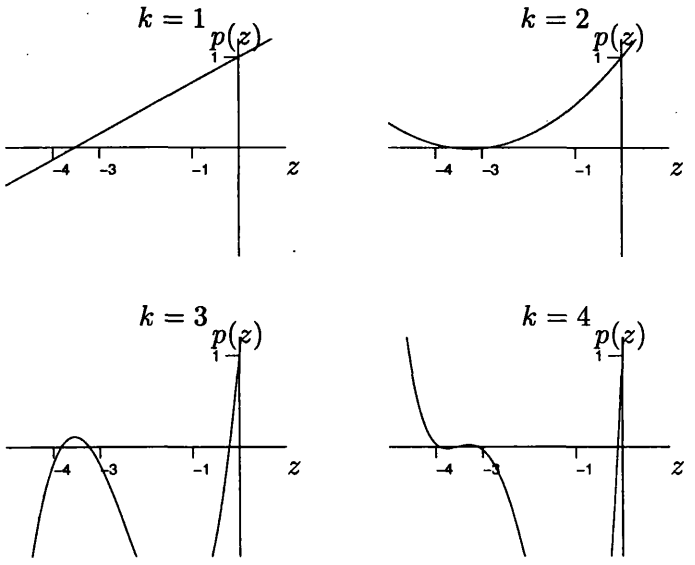


Figure 3-8: The GMRES polynomials for `A1 x1 = b1` from example 3.3

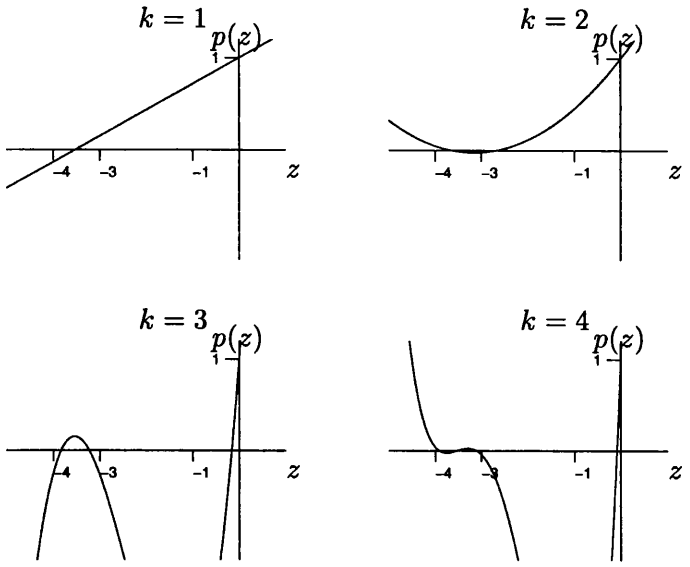


Figure 3-9: The GMRES polynomials for `A0 x0 = b0` from Example 3.3

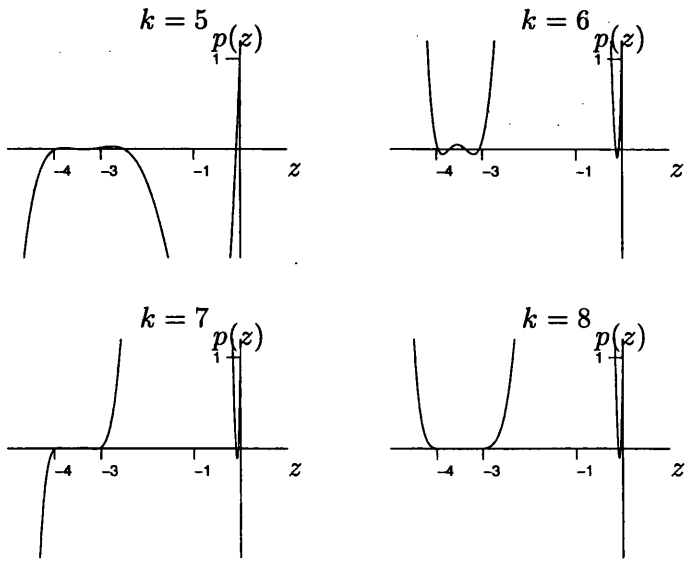


Figure 3-10: The GMRES polynomials for `A0 x0 = b0` from Example 3.3

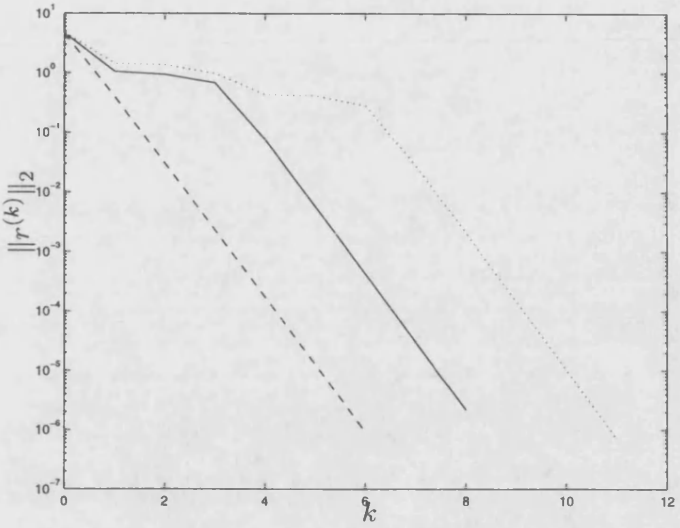


Figure 3-11: Convergence of GMRES for the matrices A0 (dotted line), A1 (solid line) and A2 (dashed line) from Example 3.3

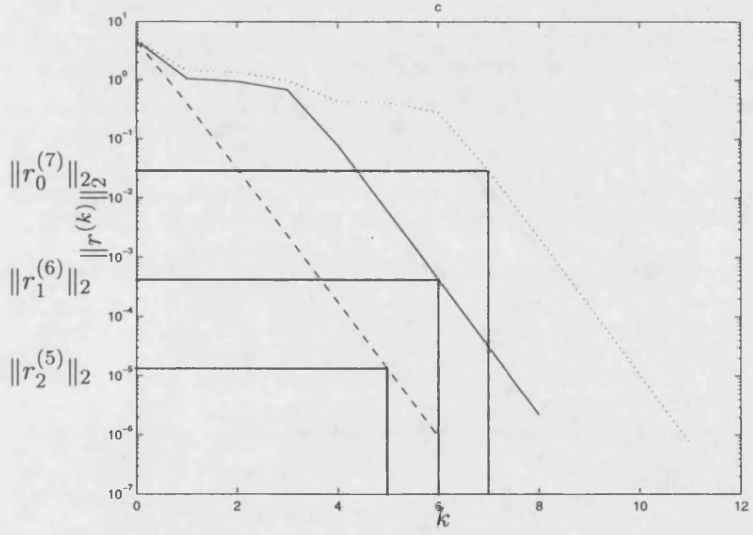


Figure 3-12: Convergence of GMRES for the matrices A0 (dotted line), A1 (solid line) and A2 (dashed line) from Example 3.3

k	$\rho_0^{(k)}$	$\rho_1^{(k)}$
1	0.2289	0.29
2	2.3839	2.49
3	22.2392	14.16
4	33.2664	178.72
5	33.4100	1634.01
6	33.4264	2261.61
7	33.4429	2269.57

Table 3.1: Table showing the ratios ρ_0 and ρ_1 from Example 3.3

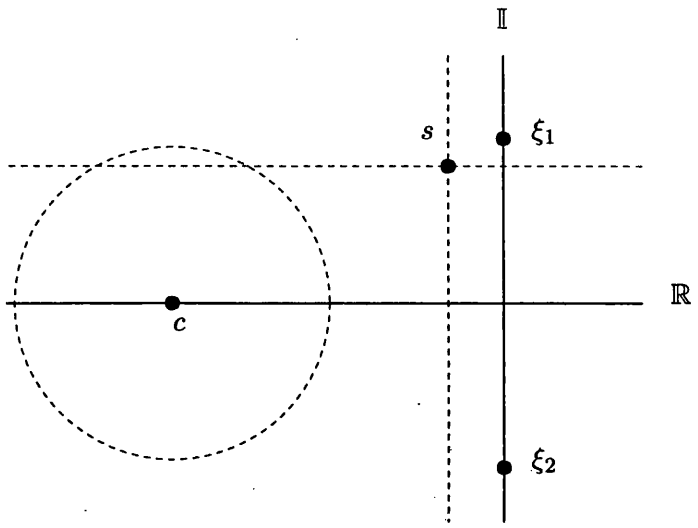


Figure 3-13: The distances between ξ_2 and c are preserved under the action of the shift. See Lemma 3.7.

Before tackling this problem we first digress slightly and compute the maximum and minimum values of the linear term

$$|1 - z/(\xi_2 - s)|$$

over the disk $D(c - s, r)$ where $c, r \in \mathbb{R}$ and $s \in \mathbb{C}$. This is a generalisation of the problem in Application 2, and is illustrated in Figure 3-13.

Lemma 3.7

Let $c, r \in \mathbb{R}$ and let $\xi_1 = x + yi \in \mathbb{C}$ where $|c - \xi_1| > r$. Let $\xi_2 = \bar{\xi_1}$ and let s be

some complex shift. Then

$$\max_{z\in D(c-s,r)} |1 - z/(\xi_2 - s)| = \frac{\sqrt{(x-c)^2+y^2}+r}{|\xi_2 - s|}$$

and

$$\min_{z\in D(c-s,r)} |1 - z/(\xi_2 - s)| = \frac{\sqrt{(x-c)^2+y^2}}{|\xi_2 - s|}.$$

Proof

We first observe that

$$\begin{aligned}\max_{z\in D(c-s,r)} |1 - z/(\xi_2 - s)| &= \left|\frac{1}{\xi_2 - s}\right| \max_{z\in D(c-s,r)} |(\xi_2 - s) - z| \\ &= \left|\frac{1}{\xi_2 - s}\right| \max_{z\in D(c,r)} |(\xi_2 - s) - (z - s)| \\ &= \left|\frac{1}{\xi_2 - s}\right| \max_{z\in D(c,r)} |\xi_2 - z|.\end{aligned}$$

Similarly

$$\min_{z\in D(c-s,r)} |1 - z/(\xi_2 - s)| = \left|\frac{1}{\xi_2 - s}\right| \min_{z\in D(c,r)} |\xi_2 - z|.$$

Now

$$\begin{aligned}\max_{z\in D(c,r)} |\xi_2 - z| &= |\xi_2 - c| + r \\ &= \sqrt{(x-c)^2+y^2}+r\end{aligned}$$

and

$$\begin{aligned}\min_{z\in D(c,r)} |\xi_2 - z| &= |\xi_2 - c| - r \\ &= \sqrt{(x-c)^2+y^2}-r.\end{aligned}$$

□

Application 2. The outlying eigenvalues are complex conjugates This application is one which frequently arises; for example, at or close to a Hopf bifurcation the Jacobian matrix has outlying complex conjugate eigenvalues. Let A have the two pure imaginary outlying eigenvalues ξ_1 and $\xi_2 = \bar{\xi}_1$. Let the other eigenvalues lie in some

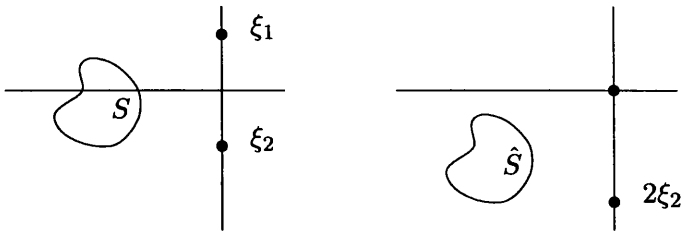


Figure 3-14: Illustration of the spectra of the shifted and unshifted matrices in Application 2.

domain S. We consider here the *shifted problem*

$$(A - sI)x = b. \tag{3.14}$$

We call $s \in \mathbb{C}$ the *shift*. Here we choose $s = \xi_1$. The transformation

$$\begin{aligned} \mathbb{C} &\rightarrow \mathbb{C} \\ z &\mapsto z - s \end{aligned}$$

has the action

$$\begin{aligned} S &\mapsto \hat{S} \\ \xi_1 &\mapsto 0 \\ \xi_2 &\mapsto 2\xi_2. \end{aligned}$$

This is illustrated in Figure 3-14.

Immediately we note that the system (3.14) is a *singular* system. The application of GMRES to singular systems is discussed in Brown and Walker [11]. We briefly consider the application of GMRES to singular systems in Section 3.4. In some cases GMRES can be used to solve singular systems. However, in this case we will not attempt to solve the singular system directly.

With our previous notation

$$\begin{aligned} S_2 &= \hat{S} \\ S_1 &= \hat{S} \cup \{2\xi_2\} \\ S_0 &= \hat{S} \cup \{0, 2\xi_2\}. \end{aligned}$$

Let $p_k^\star$ (or equivalently $q_k^{(2)}$) be the minimax polynomial for S_2. We do not assume any knowledge of the minimax polynomial $p_k^\star$, and for general domains we do not expect to be able to find such a polynomial. The pseudo-minimax polynomial $q_{k+1}^{(1)}(z)$ for S_1 is given by

$$(1 - z/(2\xi_2))p_k^\star(z).$$

We do not consider the pseudo-minimax polynomial $q_{k+2}^{(0)}$ for S_0. In this case such a polynomial does not exist; if $q_{k+2}^{(0)}$ were to exist it would satisfy $q_{k+2}^{(0)}(0) = 1$ and $q_{k+2}^{(0)}(0) = 0$.

Let $A_0 = A - \xi_1 I$. We *remove* the zero eigenvalue of A_0 to obtain the intermediate matrix A_1. A_1 has the single outlying eigenvalue $2\xi_2$. Let the matrix A_2 have the spectrum of A_0 with *both* outlying eigenvalues removed.

It is helpful here to briefly summarise our notation.

- The matrix A_0 has no eigenvalues removed. We do not consider solves with A_0.
- The matrix A_1 has *one* eigenvalue removed. The spectrum of A_1 is contained within the domain S_1. The degree k pseudo-minimax polynomial for S_1 is $q_k^{(1)}$.
- The matrix A_2 has *two* eigenvalues removed. The spectrum of A_2 is contained within the domain S_2. A_2 has *no* outlying eigenvalues and thus there is a minimax polynomial for S_2. We will denote the degree k minimax polynomial for S_2 by $p_k^\star$. It is sometimes helpful to refer to the degree k minimax polynomial as $q_k^{(2)}$. This notation is consistent with the numbering of the intermediate domains.

We now compare the convergence of GMRES for solves with the matrices A_1 and

k	$\rho^{(k)}$	k	$\rho^{(k)}$
1	0.0981	6	2.5289
2	0.1299	7	2.5285
3	1.1108	8	2.5284
4	2.7833	9	2.5284
5	2.5316		

Table 3.2: Table showing the ratio $\rho^{(k)}$ from Example 3.4

A_2. In Example 3.4 we apply GMRES to a shifted solve of the form (3.14). We consider the particular case where S is a circle. In this case the minimax polynomial for S is known.

Example 3.4

In this example we compare the convergence rate of GMRES for a matrix with no outlying eigenvalues with the convergence rate of GMRES for a matrix with one outlying eigenvalue. The matrix A considered has 84 eigenvalues on the unit circle centred at -10. A also has the two outlying eigenvalues $\xi_1 = 2i$ and $\xi_2 = -2i$. A is a real matrix. We consider the shifted solve $(A - \xi_1 I)x = b$.

It is helpful to define our intermediate domains. Let

$$\begin{aligned} S_2 &= D(10 - 2i, 1) \\ S_1 &= D(10 - 2i, 1) \cup \{-4i\}. \end{aligned}$$

Let $p_k^\star$ be the minimax polynomial for S_2. The degree $k+1$ pseudo-minimax polynomial for S_1 is given by

$$(1 - z/(-4i))p_k^\star(z).$$

By Lemma 3.7 we have that

$$\begin{aligned} \max_{z\in S_2} |1 - z/(-4i)| &= \frac{\sqrt{10^2+2^2}+1}{|-4i|} \\ &\approx 2.7995 \\ \text{and} \quad \min_{z\in S_2} |1 - z/(-4i)| &= \frac{\sqrt{10^2+2^2}-1}{|-4i|} \\ &\approx 2.2995. \end{aligned}$$

Theorem 3.6 then gives the bound

$$2.2995 \cdot \max_{z\in S_2} |p^{\star}_{k-1}(z)| \leq \max_{z\in S_2} |q^{(1)}_k(z)| \leq 2.7995 \cdot \max_{z\in S_2} |p^{\star}_{k-1}(z)|.$$

If the pseudo-minimax inequality (3.7) gives a tight bound here then we expect

$$2.2995 \cdot \|r_2^{(k-1)}\|_2 \leq \|r_1^{(k)}\|_2 \leq 2.7995 \cdot \|r_2^{(k-1)}\|_2.$$

Table 3.2 displays the ratio

$$\rho^{(k)} := \frac{\|r_1^{(k+1)}\|_2}{\|r_2^{(k)}\|_2}.$$

We see that $\lim \rho^{(k)} \approx 2.5284$. This falls within the range predicted above.

Contours of the absolute value of the GMRES polynomials $\tilde{p}_2$ and $\tilde{p}_1$ are shown in Figures 3-18 and 3-17. These are the polynomials computed at convergence. We can clearly see that $\tilde{p}_1$ is small close to $-4i$. This is not true of $\tilde{p}_2$, as we expect.

The convergence history of GMRES applied to the matrices A_1 and A_2 is shown in Figure 3-16. The slopes of the lines are in agreement with our theory.

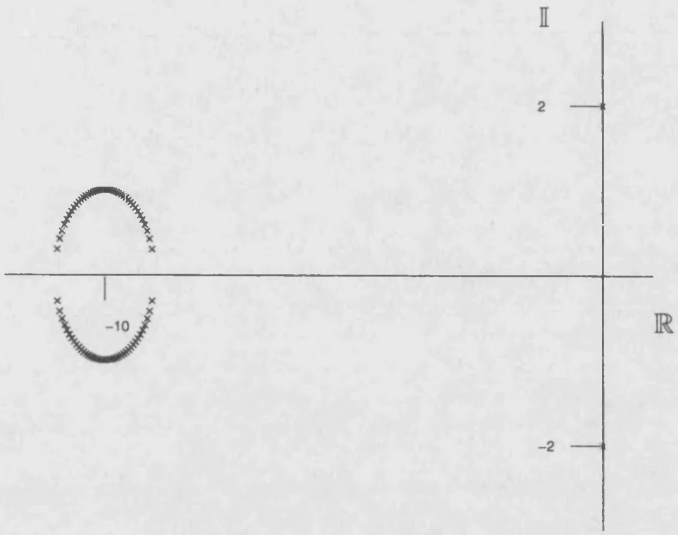


Figure 3-15: Spectrum of the matrix A_0 in Example 3.4

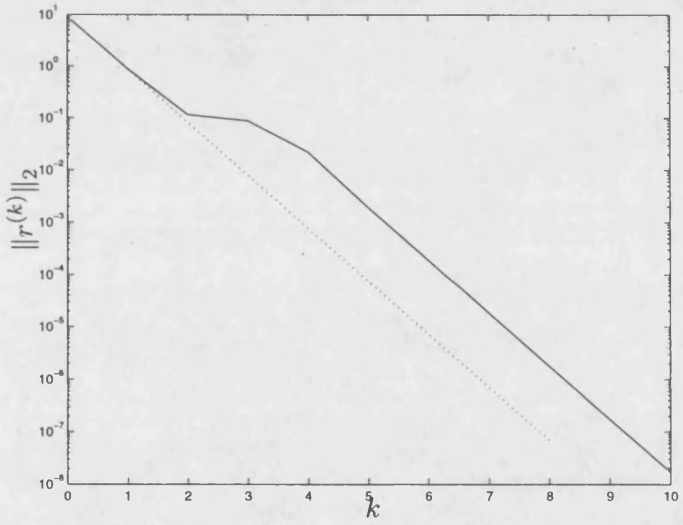


Figure 3-16: Convergence of GMRES for the matrices A_2 (dotted line), and A_1 (solid line) from Example 3.4

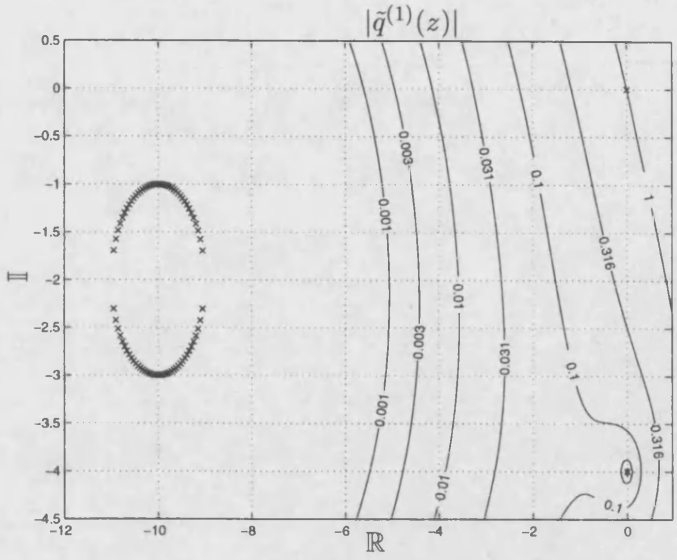


Figure 3-17: Contour plot of the modulus of the GMRES polynomial at convergence for $A_1 x = b$ in Example 3.4. The eigenvalues of A_1 are marked with crosses ($\times$).

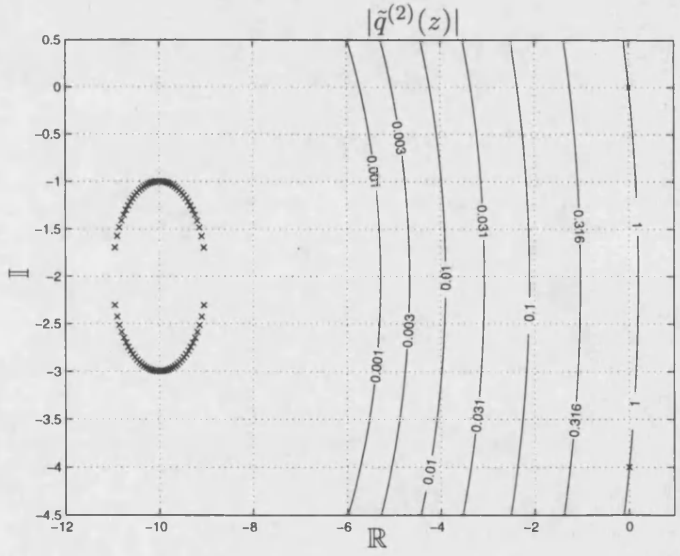


Figure 3-18: Contour plot of the modulus of the GMRES polynomial at convergence for $A_2 x = b$ in Example 3.4. The eigenvalues of A_1 are marked with crosses ($\times$).

3.4 GMRES for projected systems

Recall that A is an $n \times n$ real or complex matrix. In this section we develop a technique for constructing matrices whose spectrum is the spectrum of A with some eigenvalues removed. The technique considered uses orthogonal projections to restrict A to an appropriately chosen subspace of $\mathbb{C}^n$. We will refer to systems of equations involving such *projected matrices* as *projected systems.*

We show that GMRES may be applied to certain appropriate projected systems and that the convergence rate is governed by the eigenvalues of the projected matrix.

3.4.1 GMRES for singular systems

We will show later that, by their very nature, the projected matrices have zero eigenvalues in place of the eigenvalues of A which are removed. Thus it is appropriate to discuss the conditions under which GMRES can be used to compute solutions of a singular system

$$Ax = b \tag{3.15}$$

The application of GMRES to such a system is discussed fully in Brown and Walker [11], but we will give an overview here.

We begin with some definitions.

Definition 3.8

We say that the system (3.15) is consistent *if* $b \in \mathcal{R}(A)$.

GMRES is said to breakdown *at the kth step if* $\dim(A\mathcal{K}_k(A, r^{(0)})) \neq k$.

There are a number of types of breakdown. We will not discuss them here.

The following theorem states that for certain classes of singular matrix GMRES will compute a least squares solution for all right hand sides b and all initial guess vectors $x^{(0)}$.

Theorem 3.9 (Brown and Walker [11, Theorem 2.4])

GMRES determines a least squares solution of (3.15) for all b and $x^{(0)}$ if and only if $\mathcal{N}(A) = \mathcal{N}(A^T)$. Furthermore, if (3.15) is consistent and $x^{(0)} \in \mathcal{R}(A)$ then the solution reached is the pseudo-inverse solution.

We note here that the following are equivalent (see Brown and Walker [11]):

- $\mathcal{N}(A) = \mathcal{N}(A^T)$
- $\mathcal{N}(A) = \mathcal{R}(A)^\perp$
- $\mathcal{N}(A)^\perp$ is an invariant subspace of A.

If $\mathcal{N}(A) \neq \mathcal{N}(A^T)$ then GMRES will not produce a least squares solution *for all* $x^{(0)}$ and b. Theorem 3.10 shows that under weaker conditions GMRES will still converge for *some* $x^{(0)}$ and b.

Theorem 3.10 (Brown and Walker [11, Theorem 2.6])

Suppose that $\mathcal{N}(A) \cap \mathcal{R}(A) = \{0\}$. If (3.15) is consistent then GMRES determines a solution at some step and breaks down at the next step.

3.4.2 Theory for projected solves

Let $\mathcal{F}$ and $\mathcal{G}$ be orthogonal subspaces of $\mathbb{C}^n$ with $\mathbb{C}^n = \mathcal{F} \oplus \mathcal{G}$. Now let $\mathcal{P} : \mathbb{C}^n \to \mathcal{F}$ and $\mathcal{Q} : \mathbb{C}^n \to \mathcal{G}$ be orthogonal projections onto $\mathcal{F}$ and $\mathcal{G}$ respectively, with $\mathcal{Q} = I - \mathcal{P}$.

With these projections the n dimensional system (3.15) is equivalent to the block system

$$\begin{bmatrix} \mathcal{P}A\mathcal{P} & \mathcal{P}A\mathcal{Q} \\ \mathcal{Q}A\mathcal{P} & \mathcal{Q}A\mathcal{Q} \end{bmatrix} \begin{bmatrix} \mathcal{P}x \\ \mathcal{Q}x \end{bmatrix} = \begin{bmatrix} \mathcal{P}b \\ \mathcal{Q}b \end{bmatrix}. \tag{3.16}$$

The following example illustrates our approach in studying projected systems.

Example 3.5

Suppose that $\mathcal{F}$ and $\mathcal{G}$ are invariant under A. Then $\mathcal{Q}A\mathcal{P} = 0$, $\mathcal{P}A\mathcal{Q} = 0$, and

(3.16) becomes the block diagonal system

$$\begin{bmatrix} \mathcal{P}A\mathcal{P} & 0 \\ 0 & \mathcal{Q}A\mathcal{Q} \end{bmatrix} \begin{bmatrix} \mathcal{P}x \\ \mathcal{Q}x \end{bmatrix} = \begin{bmatrix} \mathcal{P}b \\ \mathcal{Q}b \end{bmatrix}.$$

In the special case that $b \in \mathcal{G}$, then $\mathcal{P}b = 0$, $\mathcal{Q}b = b$, and we have that

$$\begin{bmatrix} \mathcal{P}A\mathcal{P} & 0 \\ 0 & \mathcal{Q}A\mathcal{Q} \end{bmatrix} \begin{bmatrix} \mathcal{P}x \\ \mathcal{Q}x \end{bmatrix} = \begin{bmatrix} 0 \\ \mathcal{Q}b \end{bmatrix}.$$

It follows that $\mathcal{Q}A\mathcal{Q}\ x = b$.

Suppose that x is the unique solution of $\mathcal{Q}A\mathcal{Q}\ x = b$, $x \in \mathcal{G}$. Then we see that x is the unique solution of $Ax = b$, $x \in \mathcal{G}$.

The situation in the case where $\mathcal{G}$ is *not* invariant under A is similar. In this case the block system (3.16) becomes the block upper triangular system

$$\begin{bmatrix} \mathcal{P}A\mathcal{P} & \mathcal{P}A\mathcal{Q} \\ 0 & \mathcal{Q}A\mathcal{Q} \end{bmatrix} \begin{bmatrix} \mathcal{P}x \\ \mathcal{Q}x \end{bmatrix} = \begin{bmatrix} 0 \\ \mathcal{Q}b \end{bmatrix}.$$

The observation in the example above leads us to ask the following question; when does $\mathcal{Q}A\mathcal{Q}\ x = b$ have a unique solution in $\mathcal{G}$? We claim that when $\mathcal{F}$ is invariant the null space of $\mathcal{Q}A\mathcal{Q}$ is $\mathcal{F}$. Now the restriction of $\mathcal{Q}A\mathcal{Q}$ to $\mathcal{G}$ is nonsingular and so the solution in $\mathcal{G}$ of $\mathcal{Q}A\mathcal{Q}\ x = b$ is unique. The claim is proved in the following lemma.

Lemma 3.11

Suppose that the subspace $\mathcal{F}$ is invariant under A and contains the null space of A. Then

$$\mathcal{N}(\mathcal{Q}A\mathcal{Q}) = \mathcal{N}(\mathcal{Q}).$$

Proof Suppose the columns of the orthonormal matrices Z and W span $\mathcal{F}$ and $\mathcal{G}$ respectively. Then the projections $\mathcal{P}$ and $\mathcal{Q}$ may be represented by ZZ^H and WW^H respectively and $W^HZ = 0$.

The matrix $Y = [Z, W]$ is an orthonormal matrix and so the matrices A and $Y^H AY$ have the same spectrum. We now note that $W^H AZ = 0$ since $\mathcal{F}$ is invariant under A, and thus

$$Y^H AY = \begin{bmatrix} Z^H AZ & Z^H AW \\ 0 & W^H AW. \end{bmatrix}$$

which is block upper triangular. Thus $\Lambda(A) = \Lambda(Z^H AZ) \cup \Lambda(W^H AW)$. Since the null space of A is contained in $\mathcal{F}$, the zero eigenvalues of $Y^H AY$ arise as zero eigenvalues of the block $Z^H AZ$ and the remaining eigenvalues are nonzero. In particular the block $W^H AW$ is nonsingular. It follows that $\mathcal{Q}A\mathcal{Q}\, x = 0$ if and only if $\mathcal{Q}x = 0$. □

The component in $\mathcal{G}$ of the solution x of $Ax = b$ may be uniquely determined by solving in $\mathcal{G}$ the projected system $\mathcal{Q}A\mathcal{Q}\, q = \mathcal{Q}b$. We have the following lemma on this projected system.

Lemma 3.12

Suppose that $\mathcal{F}$ is invariant under A and contains the null space of A. Then if $b \in \mathcal{G}$ the projected system

$$\mathcal{Q}A\mathcal{Q}\, q = b \tag{3.17}$$

has a unique solution in $\mathcal{G}$.

Proof By the Dimension Theorem and Lemma 3.11 we have $\mathcal{R}(\mathcal{Q}A\mathcal{Q}) = \mathcal{G}$. Then $\mathcal{Q}A\mathcal{Q}$ is bijective from $\mathcal{G}$ to $\mathcal{G}$. □

3.4.3 GMRES for projected solves

We now consider the application of GMRES to solve the projected system (3.17). When $b \in \mathcal{G}$ we have that, with an appropriate initial guess vector, GMRES will compute a solution for (3.17).

Theorem 3.13

Suppose that $\mathcal{F}$ is invariant under A and contains the null space of A. If $b \in \mathcal{G}$, and the initial guess vector $x^{(0)} \in \mathcal{G}$, then GMRES will determine a solution of (3.17) at some step and breakdown at the next step.

Proof

By Lemma 3.11 we have $\mathcal{N}(\mathcal{Q}A\mathcal{Q}) = \mathcal{N}(\mathcal{Q}) = \mathcal{F}$. Since $\mathcal{R}(\mathcal{Q}A\mathcal{Q}) = \mathcal{G}$ it follows that $\mathcal{N}(\mathcal{Q}A\mathcal{Q}) \cap \mathcal{R}(\mathcal{Q}A\mathcal{Q}) = \{0\}$.

Also, $b \in \mathcal{G} = \mathcal{R}(\mathcal{Q}A\mathcal{Q})$, that is, the system (3.17) is consistent. The result follows by Theorem 3.10. □

We now consider the projected system (3.17) in the special case where $\mathcal{F}$ arises as an eigenspace of A. This special case arises, for example, in Chapters 4 and 5 where we wish to solve the projected system in place of $Ax = b$ to reduce computational costs.

Example 3.6

Suppose that A is non-normal, with non-orthogonal eigenvectors x_1, x_2, x_3 illustrated in Figure 3-19. Suppose that the eigenvalue λ_1 corresponding to the eigenvector x_1 is zero, but that the remaining eigenvalues are nonzero.

It is natural to consider (3.17) with:

(i) $\mathcal{F} = \langle x_1 \rangle$. This space is invariant under A and contains the null space of A.

(ii) $\mathcal{G} = \mathcal{F}^{\perp}$. This space is not invariant under A but is orthogonal to $\mathcal{F}$.

One might also consider the choice $\mathcal{G} = \langle x_2, x_3 \rangle$ which is represented in Figure 3-19 by a plane. This space is invariant under A but is not orthogonal to $\mathcal{F}$. In the practical applications considered in Chapters 4 and 5 we have available only a small number of the eigenvectors of A—we are not able to compute the space spanned by the remaining eigenvectors.

Let A have eigenvalues $\lambda_1, \ldots, \lambda_n$ which are ordered by absolute magnitude, and which have corresponding normalised eigenvectors $x_1, \ldots, x_n$. Let $m \in \{1, \ldots, n\}$ and suppose that there are not more than m zero eigenvalues, that is, that $\lambda_{m+1}, \ldots, \lambda_n \neq 0$. Let $\mathcal{F} = \langle x_1, \ldots, x_m \rangle$. Clearly $\mathcal{F}$ is invariant under A and contains the null space of A.

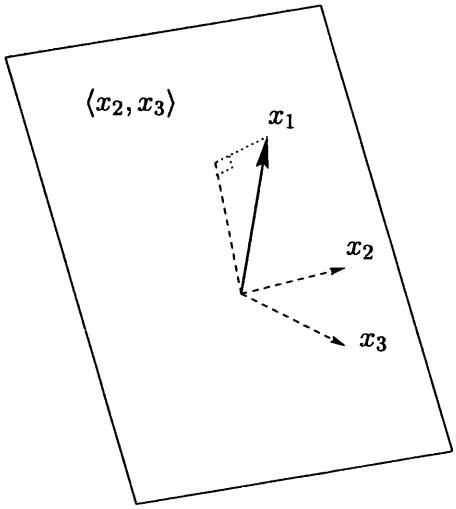


Figure 3-19: This figure illustrates the eigenvectors of the non-normal matrix described in Example 3.6.

With $\mathcal{G} = \mathcal{F}^{\perp}$,

$$\mathbb{C}^n = \mathcal{G} \oplus \mathcal{F}.$$

Note that $\mathcal{G}$ need not be invariant under A.

These choices for $\mathcal{F}$ and $\mathcal{G}$ satisfy the conditions of Lemma 3.12 and Theorem 3.13, and thus if $b \in \mathcal{G}$ then the projected system (3.17) has a unique solution in $\mathcal{G}$ which may be computed using GMRES. With these choices of $\mathcal{F}$ and $\mathcal{G}$ it is also possible to describe the eigenvalues and eigenvectors of QAQ.

Theorem 3.14

Let $\mathcal{F}$ and $\mathcal{G}$ be defined as above. Then the matrix QAQ has m eigenvalues at zero with corresponding eigenvectors $x_1, \ldots, x_m$. The remaining eigenvalues of QAQ are $\lambda_{m+1}, \ldots, \lambda_n$ with corresponding eigenvectors $Qx_{m+1}, \ldots, Qx_n$.

Proof

We first show that the vectors $x_1, \ldots, x_m$ are null vectors of $\mathcal{Q}A\mathcal{Q}$. For $i = 1, \ldots, m$ we have that

$$\begin{aligned} \mathcal{Q}A\mathcal{Q}x_i &= \mathcal{Q}A(I - \mathcal{P})x_i \\ &= \mathcal{Q}A \cdot 0 \\ &= 0. \end{aligned}$$

It remains to show that $\mathcal{Q}A\mathcal{Q}(\mathcal{Q}x_i) = \lambda_i(\mathcal{Q}x_i)$ for $i = m+1, \ldots, n$. For $i = m+1, \ldots, n$ we have that

$$\begin{aligned} \mathcal{Q}A\mathcal{Q}(\mathcal{Q}x_i) &= \mathcal{Q}A\mathcal{Q}x_i \\ &= \mathcal{Q}A(I - \mathcal{P})x_i \\ &= \mathcal{Q}Ax_i - \mathcal{Q}A\mathcal{P}x_i \\ &= \mathcal{Q}\lambda_i x_i - \mathcal{Q}A\mathcal{P}x_i \\ &= \lambda_i \mathcal{Q}x_i - \mathcal{Q}A\mathcal{P}x_i. \end{aligned}$$

Now, $\mathcal{P}x_i$ lies in the invariant subspace $\mathcal{F}$. Thus $A\mathcal{P}x_i \in \mathcal{F}$ and so $\mathcal{Q}A\mathcal{P}x_i = 0$. It follows that $\mathcal{Q}A\mathcal{Q}(\mathcal{Q}x_i) = \lambda_i(\mathcal{Q}x_i)$. □

We are now able to state the following important theorem.

Theorem 3.15

Let $\mathcal{F}$ and $\mathcal{G}$ be defined as above. If $b \in \mathcal{G}$ then the projected system (3.17)

$$\mathcal{Q}A\mathcal{Q}\ x = b$$

has the following properties:

(i) (3.17) has a unique solution in $\mathcal{G}$.

(ii) If the initial guess vector $x^{(0)}$ lies in $\mathcal{G}$ then GMRES will compute at some step a least squares solution for (3.17), and breaks down at the next step.

(iii) If the initial guess vector $x^{(0)}$ lies in $\mathcal{G}$ then the rate of convergence of GMRES for (3.17) has no dependence on the eigenvalues $\lambda_1, \ldots, \lambda_m$.

Proof

(i) This is Lemma 3.12.

(ii) This is Theorem 3.13.

(iii) The eigenvectors of $\mathcal{Q}A\mathcal{Q}$ are given by Theorem 3.14. Observe that the eigenvectors $\mathcal{Q}x_{m+1}, \ldots, \mathcal{Q}x_n$ of $\mathcal{Q}A\mathcal{Q}$ are orthogonal to the eigenvectors $x_1, \ldots, x_m$. It follows that the initial residual $r^{(0)} = b - Ax^{(0)}$ is in $\mathcal{G}$ and so has no component in the vectors $x_1, \ldots, x_m$. The result follows by Lemma 3.2.

□

3.4.4 Constructing projections

We now consider how to construct the projections $\mathcal{P}$ and $\mathcal{Q}$ onto the subspaces $\mathcal{F}$ and $\mathcal{G}$ described above. Recall that $\mathcal{F}$ is the space spanned by the first m eigenvectors $x_1, \ldots, x_m$ of A. We construct the orthogonal projection $\mathcal{P}$ onto $\mathcal{F}$ as follows:

Let $\hat{X} = [x_1, \ldots, x_m]$ and let ZU be a QR decomposition of $\hat{X}$. Then $\mathcal{R}(\hat{X}) = \mathcal{R}(Z)$ and Z is an orthonormal matrix. Define $\mathcal{P}$ by

$$\mathcal{P} = ZZ^H. \tag{3.18}$$

$\mathcal{P}$ is an orthogonal projection onto $\langle x_1, \ldots, x_m \rangle$.

$\mathcal{Q} := I - \mathcal{P}$ is an orthogonal projection onto $\mathcal{G}$.

Remark

Since $\langle x_1, \ldots, x_m \rangle$ is invariant under A there exist orthonormal matrices S with $\mathcal{R}(S) = \langle x_1, \ldots, x_m \rangle$ and with the property that

$$AS = ST \tag{3.19}$$

where T is an $m \times m$ upper triangular matrix. The columns of S are called *Schur vectors* of A. Equation (3.19) is called a *Schur decomposition* of A. Schur decompositions are not unique.

3.4.5 Projections from approximate eigenvectors

In Chapters 4 and and 5 we wish to solve projected systems of the form (3.17)

$$\mathcal{Q}A\mathcal{Q}\ q = b$$

in place of the system $Ax = b$. In the context of our practical application we have not yet computed any of the eigenvectors of A and so cannot compute the projections $\mathcal{P}$ and $\mathcal{Q}$ described above.

However, we may assume that we have the normalised approximations $\hat{x}_1, \ldots, \hat{x}_m$ to the eigenvectors $x_1, \ldots, x_m$ of A. We now consider the solution of projected systems which are obtained using projections onto $\mathcal{F} = \langle \hat{x}_1, \ldots, \hat{x}_m \rangle$ and $\mathcal{G} = \mathcal{F}^{\perp}$.

We assume that $\hat{x}_1, \ldots, \hat{x}_m$ arise as *Ritz vectors* and that each Ritz vector $\hat{x}_i$ has associated with it a *Ritz value* θ_i. Recall that the Ritz value θ_i approximates the eigenvalue λ_i.

Each Ritz pair $(\hat{x}_i, \theta_i)$ has a residual

$$r_i = A\hat{x}_i - \theta_i \hat{x}_i$$

for the eigenvalue problem $Ax = \lambda x$. Each residual r_i is orthogonal to all of the Ritz vectors. Note that the residuals r_i of the eigenvalue problem are not related to the residual $r^{(j)}$ of the approximate solution $x^{(j)}$ for the system $Ax = b$.

We may construct the projections $\mathcal{P}$ and $\mathcal{Q}$ onto $\mathcal{F}$ and $\mathcal{G}$ in the same way as in the previous section. We will refer to these projections, which are constructed from approximate eigenvectors, as *approximate projections.* The approximate projections share many of the properties of the original projections when used in projected solves. In particular, we see that $n-m$ of the eigenvalues of the projected matrix $\mathcal{Q}A\mathcal{Q}$ remain

close to eigenvalues of A. The remaining eigenvalues are at zero. This is given by the following theorem and its corollary.

Theorem 3.16

Let $\mathcal{F}$, $\mathcal{G}$, $\mathcal{P}$ and $\mathcal{Q}$ be defined as above. Then

(i) the matrix $\mathcal{Q}A\mathcal{Q}$ has m eigenvalues at zero with corresponding eigenvectors $\hat{x}_1, \ldots, \hat{x}_m$.

(ii) the pairs $(\lambda_j, \mathcal{Q}x_j)$ for $j = m+1, \ldots, n$ are approximate eigenpairs of $\mathcal{Q}A\mathcal{Q}$ with residuals $-RU^{-1}Z^H x_j$,

where ZU is an incomplete QR decomposition of $\hat{X} = [\hat{x}_1, \ldots, \hat{x}_m]$ and $R = [r_1, \ldots, r_m]$.

Proof

It is immediate that the vectors $\hat{x}_1, \ldots, \hat{x}_m$ are eigenvectors of $\mathcal{Q}A\mathcal{Q}$ with zero eigenvalues.

From (3.18) we have that $\hat{X} = [\hat{x}_1, \ldots, \hat{x}_m]$ and the QR decomposition $\hat{X} = ZU$ of $\hat{X}$. It is convenient to write $R = [r_1, \ldots, r_m]$ and $\Lambda = \text{diag}_{i=1,\ldots,m}(\theta_i)$. Then

$$\begin{aligned} A\hat{X} &= \hat{X}\Lambda + R, \\ \text{that is } AZU &= ZU\Lambda + R \\ \text{so that } AZ &= ZU\Lambda U^{-1} + RU^{-1}. \end{aligned}$$

Let $m+1 \leq i \leq n$. Then

$$\begin{aligned} \mathcal{Q}A\mathcal{Q}(\mathcal{Q}x_i) &= \mathcal{Q}A\mathcal{Q}x_i \\ &= \mathcal{Q}A(x_i - \mathcal{P}x_i) \\ &= \mathcal{Q}Ax_i - \mathcal{Q}A\mathcal{P}x_i \\ &= \mathcal{Q}Ax_i - \mathcal{Q}A(ZZ^H x_i). \end{aligned}$$

For convenience we will write $\epsilon_i = Z^H x_i$. Then

$$\begin{aligned} \mathcal{Q}A\mathcal{Q}(\mathcal{Q}x_i) &= \mathcal{Q}Ax_i - \mathcal{Q}\ AZ\ \epsilon_i \\ &= \mathcal{Q}Ax_i - \mathcal{Q}(ZU\Lambda U^{-1} + RU^{-1})\epsilon_i \\ &= \mathcal{Q}Ax_i - \mathcal{Q}ZU\Lambda U^{-1}\epsilon_i - \mathcal{Q}RU^{-1}\epsilon_i. \end{aligned}$$

Since $\mathcal{P}Z = Z$ it follows that $\mathcal{Q}Z = 0$.

Since $R \perp \langle \hat{x}_1, \ldots, \hat{x}_m \rangle$ we have that $\mathcal{Q}R = R$. Thus

$$\begin{aligned} \mathcal{Q}A\mathcal{Q}(\mathcal{Q}x_i) &= \mathcal{Q}Ax_i - RU^{-1}\epsilon_i \\ &= \mathcal{Q}\lambda_i x_i - RU^{-1}\epsilon_i \\ &= \lambda_i \mathcal{Q}x_i - RU^{-1}\epsilon_i, \end{aligned}$$

that is $\mathcal{Q}A\mathcal{Q}(\mathcal{Q}x_i) - \lambda_i(\mathcal{Q}x_i) = -RU^{-1}\epsilon_i$. □

Let λ_j be an eigenvalue of A. We shall denote by η_j the eigenvalue of $\mathcal{Q}A\mathcal{Q}$ corresponding to λ_j. Theorem 3.16 and the Bauer-Fike Theorem [56, Theorem 3.6] allow us to bound the distance between each eigenvalue λ_j and its corresponding eigenvalue η_j of $\mathcal{Q}A\mathcal{Q}$. This bound is given in the following corollary.

Corollary 3.17

Let $m+1 \leq j \leq n$. Under the conditions of Theorem 3.16

$$\begin{aligned} |\eta_j - \lambda_j| &\leq \kappa_2(Y)\|RU^{-1}Z^H x_j\|_2 \\ &\leq \kappa_2(Y)\|R\|_2\|U^{-1}\|_2\|X^H x_j\|_2 \end{aligned} \tag{3.20}$$

where $Y = [\hat{x}_1, \ldots, \hat{x}_m, \mathcal{Q}x_{m+1}, \ldots, \mathcal{Q}x_n]$.

Proof

This is an immediate corollary of the Bauer-Fike Theorem [56, Theorem 3.6] and Theorem 3.16. □

We have shown that the distance between the eigenvalue η_j of $\mathcal{Q}A\mathcal{Q}$ and the eigen-

value λ_j of A is proportional to both the magnitudes of the residuals of the approximate eigenpairs and the component of x_j in the directions of the approximate eigenpairs. When these are small we see then the distance between η_j and λ_j will be small.

In the following example we look at the eigenvalues of a matrix A and compare them with the eigenvalues of a projected matrix QAQ.

Example 3.7

Recall the matrix A of Example 3.4. A is an 86×86 real matrix and is similar to the block diagonal matrix whose diagonal consists of 43 two by two blocks of the form

$$B = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}.$$

Each block B has eigenvalues $a \pm bi$. A is non-normal, but each pair of complex conjugate eigenvectors of A is orthogonal to any other pair of complex conjugate eigenvectors of A.

In this example we compare the eigenvalues of A with those of QAQ, where $\mathcal{P}$ is an orthogonal projection and $Q = I - \mathcal{P}$.

Recall that A has the pair of pure imaginary conjugate eigenvalues $\lambda_{1,2} = \pm 2i$. We construct $\mathcal{P}$ from approximations to the eigenvectors corresponding to the eigenvalues $\pm 2i$. We form the approximate eigenvectors by perturbing the eigenvectors x_1 and x_2. The perturbed eigenvectors have residuals r_1 and r_2 and

$$\|r_1\|_2 = \|r_2\|_2 = 0.3575.$$

Figures 3-20 and 3-21 show the eigenvalues of A and QAQ. Note that $\pm 2i$ are *not* eigenvalues of QAQ. The eigenvalues $\pm 2i$ have been transformed to zero eigenvalues of QAQ. The other eigenvalues are slightly perturbed. Figure 3-21 shows the main part of the spectrum in more detail.

We now state and prove an analogue of Theorem 3.15 for the case with inexact projections.

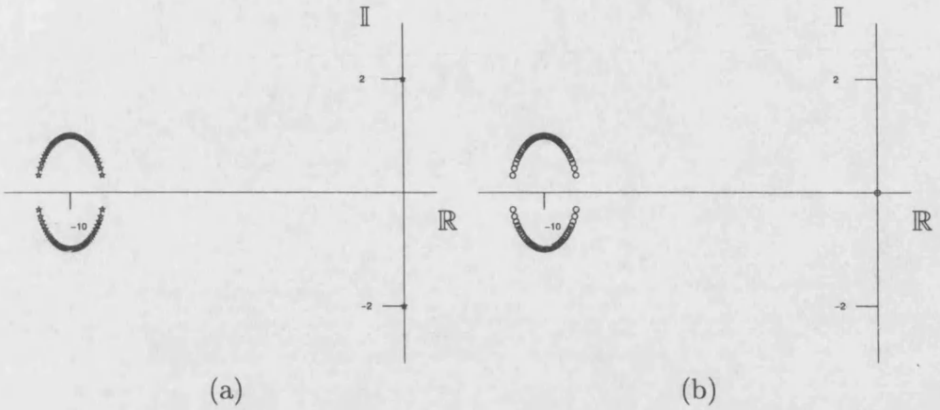


Figure 3-20: The eigenvalues of (a) the matrix A, and (b) the matrix $\mathcal{Q}A\mathcal{Q}$.

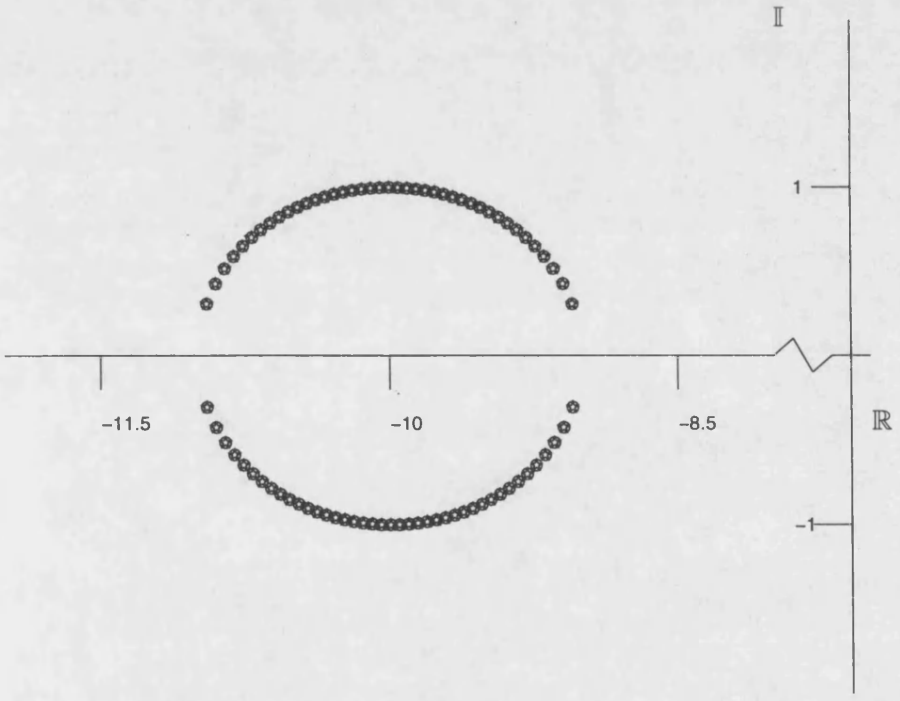


Figure 3-21: The eigenvalues of A ($\star$) and $\mathcal{Q}A\mathcal{Q}$ ($\circ$) in more detail.

Theorem 3.18

Let $\mathcal{F}$,$\mathcal{G}$,$\mathcal{P}$ and $\mathcal{Q}$ be defined as above, and suppose that $b \in \mathcal{G}$. Then for $\hat{x}_1, \ldots, \hat{x}_m$ sufficiently good approximations to the eigenvectors $x_1, \ldots, x_m$ the projected system (3.17)

$$\mathcal{Q}A\mathcal{Q}\, x = b$$

has the following properties:

(i) (3.17) has a unique solution $x \in \mathcal{G}$,

(ii) If the initial guess vector $x^{(0)}$ lies in $\mathcal{G}$ then GMRES will compute at some step a least squares solution for (3.17) and break down at the next step,

(iii) If the initial guess vector $x^{(0)}$ lies in $\mathcal{G}$ then the rate of convergence of GMRES for (3.17) has no dependence on the zero eigenvalues $\eta_1, \ldots, \eta_m$ of $\mathcal{Q}A\mathcal{Q}$ corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_m$ of A.

Proof

(i) We follow closely the argument used to prove Lemma 3.11.

Suppose the columns of the orthonormal matrices $\hat{Z}$ and $\hat{W}$ span $\mathcal{F}$ and $\mathcal{G}$ respectively. Then then projections $\mathcal{P}$ and $\mathcal{Q}$ may be represented by $\hat{Z}\hat{Z}^H$ and $\hat{W}\hat{W}^H$ respectively and $\hat{W}^H\hat{Z} = 0$.

Suppose also that the columns of the orthonormal matrices Z and W span $\langle x_1, \ldots, x_m \rangle$ and $\langle x_1, \ldots, x_m \rangle^{\perp}$ respectively.

We showed in Lemma 3.11 that the matrix W^HAW is nonsingular. We now observe that the matrix $\hat{W}^HA\hat{W}$ is a perturbation of the matrix W^HAW and for $\hat{x}_1, \ldots, \hat{x}_m$ sufficiently close to $x_1, \ldots, x_m$ we have W^HAW is nonsingular. It follows that $\mathcal{Q}A\mathcal{Q}x = 0$ if and only if $\mathcal{Q}x = 0$.

(ii) By the Dimension Theorem and part (i) we have $\mathcal{R}(\mathcal{Q}A\mathcal{Q}) = \mathcal{G}$. Thus $\mathcal{N}(\mathcal{Q}A\mathcal{Q}) \cap \mathcal{R}(\mathcal{Q}A\mathcal{Q}) = \{0\}$ and the result follows by Theorem 3.10.

(iii) Let $\eta_1, \ldots, \eta_n$ denote the eigenvalues of $\mathcal{Q}A\mathcal{Q}$, where the eigenvalue η_i is related to the eigenvalue λ_i of A in the natural way. Then $\eta_1, \ldots, \eta_m$ are zero and their corresponding eigenvectors lie in $\mathcal{F}$. The remaining eigenvectors lie in $\mathcal{G}$.

Let $x^{(0)} \in \mathcal{G}$. Then the initial residual $r^{(0)}$ has no component in any of the eigenvectors of $\mathcal{Q}A\mathcal{Q}$ in $\mathcal{F}$ and the result follows by Lemma 3.2.

□

We illustrate the application of Theorem 3.18 with an example.

Example 3.8

Let A be a matrix with eigenvalues $\lambda_1, \ldots, \lambda_7$, represented by the crosses in Figure 3-22, and with corresponding eigenvectors $x_1, \ldots, x_7$. It is clear that A has an eigenvalue λ_1 very close to the origin. The results of Sections 3.2 and 3.3 show that the presence of this eigenvalue will slow down the convergence of GMRES when it is applied to linear systems involving A.

Let $\hat{x}_1$ be an approximation to x_1, and suppose that we can reformulate our problem which involves the solve $Ax = b$, in such a way that we can instead solve

$$\mathcal{Q}A\mathcal{Q}\, y = c \tag{3.21}$$

where $\mathcal{Q}$ is a projection onto the subspace $\langle \hat{x}_1 \rangle^{\perp}$

By Corollary 3.17 we have that each of the eigenvalues $\eta_2, \ldots, \eta_7$ of $\mathcal{Q}A\mathcal{Q}$ lies in a disk around the corresponding eigenvalue λ_i of A. The radius of the ith disk depends upon the residual of $\hat{x}_1$ as an eigenvector approximation to x_1, and upon the component of the eigenvector x_i in the direction of $\hat{x}_1$.

Theorem 3.18 tells us that GMRES will compute the unique solution y of the system (3.21). Theorem 3.18 also states that the convergence rate of GMRES depends upon the eigenvalues $\eta_2, \ldots, \eta_7$.

If the approximation $\hat{x}_1$ to x_1 is very good then the radii of all of the disks will be small—each η_i will be very close to λ_i, and GMRES will converge as if it were being applied to a solve with the matrix A, but with the eigenvalue λ_1 removed. The results of Section 3.3 show that this will improve the convergence rate; the residual at each

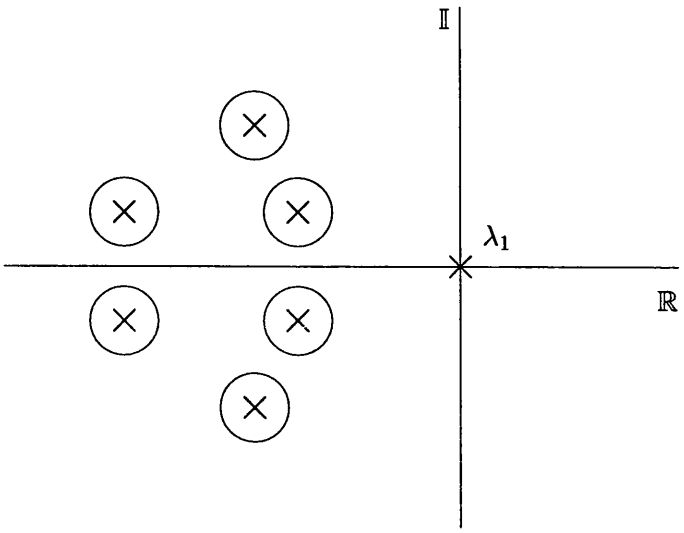


Figure 3-22: The eigenvalues of the matrix in Example 3.8.

step will be effectively on step ahead, *and* reduced by a large factor which is dependent on the distance of λ_1 from the origin, relative to the gap between λ_1 and the rest of the spectrum.

3.5 Summary

It is well known that the presence of small, outlying eigenvalues of A can reduce the convergence rate of GMRES. We have extended previous results which quantify the convergence rate for such matrices and used them in a new way to predict the improvement achieved by *removing* these small eigenvalues.

We have developed projections that, when the correct approximate eigenvectors are available, restrict A to a subspace on which it has no small eigenvalues. Results on the eigenvalues and eigenvectors of *projected matrices* are given and are used to predict the improvement in convergence rate of GMRES applied to systems with these projected matrices when compared with GMRES applied to the original system. We showed in Section 3.3 that the improvement in convergence when eigenvalues are removed can be substantial.

The projected matrices that arise in the Jacobi-Davidson method are of the type

that have been analysed is this chapter. This analysis thus provides new insight into the interaction between the Jacobi-Davidson method and GMRES which is its inner iteration. In particular we have revealed why GMRES can compute satisfactory solutions within Jacobi-Davidson more quickly than it does within Inverse Iteration.

The analysis in this chapter also shows that, when there is more than one small eigenvalue, removing just one eigenvalue does not produce significant gain—for example, see the numerical results in Example 3.3. This indicates that, when there are a number of eigenvalues close to the desired eigenvalue, solving with the projected matrix in Jacobi-Davidson provides no significant gain.

Of course *removing* all of the small eigenvalues will provide a significant gain—this motivates the following chapters.

Chapter 4

Splitting Inverse Iteration

4.1 Introduction

Is it possible to efficiently use iterative solvers to solve the near singular systems which arise when computing eigenvalues using methods based on the *Shift-Invert* transformation? In Chapter 3 we showed that iterative solvers can work efficiently on special systems derived from near singular systems. In this chapter we show how we may obtain such systems when performing Inverse Iteration and the Rayleigh Quotient Iteration. This leads us to develop a new method for computing the eigenvalues of large, sparse matrices.

We develop a simple method based on the Rayleigh Quotient Iteration which uses a combination of GMRES and a direct method to solve the systems which arise when applying the Shift-Invert transformation. When the eigenvalue distribution of the matrix is favourable this method is cheaper than the Rayleigh Quotient Iteration applied with GMRES. This method is of academic rather than practical interest but it illustrates our approach to iteratively implementing a Shift-Invert method.

An extension of this approach is used in Chapter 5 to apply iteratively the Accelerated Rayleigh Quotient Iteration. This gives a new method which we call the Iterative Accelerated Rayleigh Quotient Iteration.

The outline for this chapter is as follows. In section 4.2 we discuss the implementation of Inverse Iteration and the Rayleigh Quotient Iteration (RQI), and observe

that the shifted systems which arise are nearly singular, but only in a small number of eigendirections. On the space given by these directions we use direct solves. On the remaining space we use iterative solvers. In Section 4.3 we show how this may be done for the RQI and develop a simple method which is cheaper to implement than the RQI.

4.2 Inverse Iteration

4.2.1 Shift-Invert algorithms

Let A be a real or complex, large, square, sparse matrix. In this chapter, as in Chapter 2, we concern ourselves with the problem of computing a small number of the eigenvalues and eigenvectors of A. For simplicity we will assume that A is diagonalisable.

Recall, from Chapter 1, the *Power Method* and its variant *Inverse Iteration* (Algorithm 1.5). The Power Method and Inverse Iteration are *iterative methods* which compute a single eigenvector of a given matrix. These methods are simple to implement, and are very powerful (see Wilkinson [75, Ch. 9]).

At each step of Inverse Iteration a system of the form

$$(A - sI)y = \hat{x} \tag{4.1}$$

must be solved. The real or complex scalar s is called the *shift*. The choice of shift is important—Inverse Iteration computes the eigenvector of the matrix A whose corresponding eigenvalue is the closest eigenvalue of A to s (see Parlett [50, Sec. 4.2.2], and Householder [33, Sec. 7.4]). The approximate eigenvector $\hat{x}^{(k)}$ computed at step k of Inverse Iteration converges linearly to this eigenvector of A, with convergence factor

$$\frac{|\lambda_k - s|}{\min_{j \neq k} |\lambda_j - s|},$$

where λ_k is the closest to s of the eigenvalues $\lambda_1, \ldots, \lambda_n$ of A. When s is close to an eigenvalue of A the convergence is fast.

The system (4.1) must be solved repeatedly with different right hand sides $\hat{x}$. Thus

it is common to compute an LU factorisation of $(A - sI)$ before beginning the iteration. In this way the LU factorisation need only be computed once.

Recall that for a given matrix A the *Rayleigh Quotient* of a vector x is given by

$$\rho(x) = \frac{x^H A x}{x^H x}. \tag{4.2}$$

The Rayleigh Quotient Iteration (RQI) is an extension of Inverse Iteration which uses a different shift at each step. The shift used at step k is the Rayleigh Quotient of the current approximate eigenvector $x^{(k)}$. An implementation of the Rayleigh Quotient Iteration is given in Algorithm 1.6.

If A is normal then the Rayleigh Quotient $\rho^{(k)}$ in Algorithm 1.6 converges cubically to an eigenvalue of A (Ostrowski [47]). Furthermore, the approximate eigenvector $\hat{x}^{(k)}$ converges cubically to the corresponding eigenvector of A (see Parlett [50, §4.7]). If A is non-normal then $\hat{x}^{(k)}$ converges quadratically when A is non-defective, and linearly when A is defective. Parlett [49] describes some generalisations of the Rayleigh Quotient Iteration which improve upon these rates for non-normal matrices.

Although the Rayleigh Quotient Iteration converges more quickly than Inverse Iteration the systems solved change at each step. Consequently it is not possible to reuse a previously computed LU factorisation.

Iterative methods which solve systems such as (4.1) at each step are often called *Shift-Invert* methods.

4.2.2 Implementing Inverse Iteration

The main expense in implementing Inverse Iteration and the Rayleigh Quotient Iteration is in solving (4.1), namely $(A - sI)y = \hat{x}$. *Direct methods* are usually used for (4.1), particularly in Inverse Iteration where the LU factorisation may be reused. When A is large and sparse such direct methods are less attractive—the cost of computing an LU factorisation is typically $\mathcal{O}(n^3)$ and the factors L and U may have a less desirable sparsity structure than A. The matrices L and U may be full, in which case the cost of storing L and U may be too great. Direct solvers which perform much better for sparse

systems are available, see for example Duff and Reid [21], and Duff, Gould, Reid, and Scott [19].

Iterative methods are an alternative to direct methods. The typical cost of computing a solution for (4.1) iteratively is $\mathcal{O}(n^2)$. (Trefethen and Bau [74, Part VI] discuss in more detail the relative merits of direct and iterative methods.) If A is symmetric positive definite then one can use Conjugate Gradients (Hestenes and Skefel [32], [36]). When A is nonsymmetric *Krylov* solvers such as GMRES and BiCGSTAB may be used. These require the storage of a sequence of vectors, but can be *restarted* to limit the size of this sequence (see Saad and Schultz [59]).

Iterative methods are not commonly used for Inverse Iteration or the Rayleigh Quotient Iteration. One reason why iterative solvers are not commonly used for Inverse Iteration is that the method often stagnates—this was demonstrated for GMRES in Section 2.2 where we showed why stagnation occurs. Another reason is that if the solves are not performed to high accuracy the mapping properties of $(A-sI)^{-1}$ are not preserved (see Meerbergen [39, §2.3.4]) although when the eigenvalues of $(A - sI)$ are suitably distributed, for example concentrated away from the origin, accurate solutions can be cheaply obtained. In Inverse Iteration it is desirable to choose the shift s to be close to an eigenvalue of A, whilst in the Rayleigh Quotient Iteration the shift actually converges to an eigenvalue of A. It follows that $(A - sI)$ has an eigenvalue close to zero and we showed in Chapter 3 that Iterative solvers typically converge more slowly in this case.

4.2.3 Inverse Iteration with iterative solvers

In this chapter we consider the problem of implementing Inverse Iteration type methods (or *Shift-Invert* methods) using iterative solvers.

In the previous section we remarked that in Inverse Iteration and the Rayleigh Quotient Iteration the matrix $(A-sI)$ has an eigenvalue near zero—it is nearly singular. We now observe that $(A - sI)$ typically has only a small number of eigenvalues near zero, and so is *nearly singular in only a small number of directions.* The eigenvectors corresponding to the small eigenvalues of $(A - sI)$ span a space on which we might say

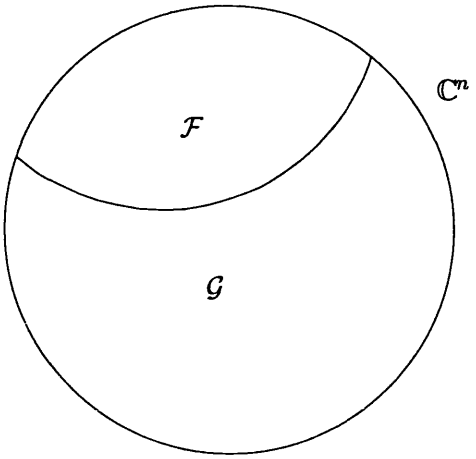


Figure 4-1: The "good" $\mathcal{G}$ and "bad" $\mathcal{F}$ spaces for $(A - sI)$.

that the restriction of $(A - sI)$ is badly conditioned. The restriction of $(A - sI)$ on the remaining space is well conditioned.

In this chapter we consider ways of replacing (4.1) by a solve over the "good space", and a solve over the "bad space". The solution y of (4.1) may then be computed by combining the solutions from both spaces. The solution on the "good space" may be computed using an iterative solver. The solution on the "bad space" may be computed using a direct solver.

It is convenient to order the eigenvalues $\lambda_1, \ldots, \lambda_n$ of A by their closeness to the shift s. With this ordering we have

$$|\lambda_1 - s| \leq |\lambda_2 - s| \leq \cdots \leq |\lambda_n - s|.$$

As usual we denote by $x_1, \ldots, x_n$ the eigenvectors corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_n$.

Suppose that m of the eigenvalues of A are close to s, that is, that m of the eigenvalues of $(A - sI)$ are small. With the above ordering these eigenvalues of A are $\lambda_1, \ldots, \lambda_m$ and their corresponding eigenvectors are $x_1, \ldots, x_m$. We say that $(A - sI)$ is "badly behaved" over the eigenspace $\langle x_1, \ldots, x_m \rangle$. It is convenient to call this space the "bad space". We know from Chapter 3 that the eigenvalues $\lambda_1, \ldots, \lambda_m$ and the eigenvectors $x_1, \ldots, x_m$ are responsible for the poor performance of iterative solvers for (4.1). We

do not, in practise, know these eigenvalues and eigenvectors. However, for small m we may have available approximations $\hat{x}_1, \ldots, \hat{x}_m$ to the eigenvectors $x_1, \ldots, x_m$. We denote by $\mathcal{F}$ the space $\langle \hat{x}_1, \ldots, \hat{x}_m \rangle$ spanned by the m approximate eigenvectors . We choose a space $\mathcal{G}$ to complement $\mathcal{F}$. It is desirable that $\mathcal{G}$ satisfies

$$\mathbb{C}^n = \mathcal{G} \oplus \mathcal{F},$$

and thus the choice of $\mathcal{G}$ is dependent on the choice of $\mathcal{F}$. To construct $\mathcal{G}$ we first find an orthogonal projection $\mathcal{P}$ onto $\mathcal{F}$. We then define the orthogonal projection $\mathcal{Q} = I - \mathcal{P}$, and let $\mathcal{G} := \mathcal{R}(\mathcal{Q})$. This choice of $\mathcal{G}$ satisfies

(i) $\mathbb{C}^n = \mathcal{G} \oplus \mathcal{F}$,

(ii) $\mathcal{G} \perp \mathcal{F}$.

We define the projection $\mathcal{P}$ onto $\mathcal{F}$ as follows, using the construction from Section 3.4.4:

> Let $\hat{X} = [\hat{x}_1, \ldots, \hat{x}_m]$ and let ZU be a QR decomposition of $\hat{X}$. Then $\mathcal{R}(\hat{X}) = \mathcal{R}(Z)$ and Z is an orthonormal matrix. Define
>
> $$\mathcal{P} = ZZ^H.$$

In Section 4.3, and in the following chapter, we present techniques which use the projections $\mathcal{P}$ and $\mathcal{Q}$ to *split* the solve $(A - sI)y = \hat{x}$ into a coupled pair of equations. These decouple to give a large system which can be solved using iterative methods, and a small system which can be solved using direct methods. The small system has dimension m. We will refer to m as the *split size*.

4.3 Splitting the Rayleigh Quotient Iteration

4.3.1 Decoupling the systems

Here we consider a split size of 1. With this choice $\mathcal{F} = \langle z \rangle$ where z is some approximation to x_1, and $\mathcal{G} = \mathcal{F}^{\perp}$. Let W be an $n \times (n-1)$ orthonormal matrix such that

$z^H W = 0$ and $W^H z = 0$. Then the columns of W span $\mathcal{G}$ and we have the projections $\mathcal{P} = zz^H$ and $\mathcal{Q} = WW^H$ onto $\mathcal{F}$ and $\mathcal{G}$ respectively. Consider the system

$$(A - sI)y = \hat{x}_1.$$

Then $[z, W]^H (A - sI)y = [z, W]^H \hat{x}_1$ and writing $p = z^H y$, $q = W^H y$, we have

$$\begin{bmatrix} z^H(A-sI)z & z^H(A-sI)W \\ W^H(A-sI)z & W^H(A-sI)W \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} z^H\hat{x}_1 \\ W^H\hat{x}_1 \end{bmatrix}. \tag{4.3}$$

Note that here s is not assumed to be the Rayleigh Quotient of z. However, z approximates an eigenvector of A and s is an approximation to the eigenvalue corresponding to this eigenvector.

The following example illustrates our approach in the simple case when $n = 2$, $m = 1$.

Example 4.1

We make the reasonable assumption that, for small ϵ, $z^H(A-sI)z = \mathcal{O}(\epsilon^2)$, $z^H(A-sI)W = \mathcal{O}(\epsilon)$, $W^H(A-sI)z = \mathcal{O}(\epsilon)$, $W^H(A-sI)W \gg 1$, and also that $z^H\hat{x}_1 = \mathcal{O}(1)$, $W^H\hat{x}_1 = \mathcal{O}(\epsilon)$. With these assumptions (4.3) becomes

$$\begin{bmatrix} \epsilon^2 & \epsilon \\ \epsilon & K \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} \alpha \\ \epsilon \end{bmatrix}. \tag{4.4}$$

This has solution

$$\begin{aligned} \begin{bmatrix} p \\ q \end{bmatrix} &= \frac{1}{\epsilon^2 K - \epsilon^2} \begin{bmatrix} K\alpha - \epsilon^2 \\ \epsilon^3 - \epsilon\alpha \end{bmatrix} \\ &\approx \frac{1}{\epsilon^2 K} \begin{bmatrix} K\alpha \\ -\epsilon\alpha \end{bmatrix}, \end{aligned}$$

assuming $K \gg 1$ and neglecting $\mathcal{O}(\epsilon^2)$ and smaller terms. But $(1/\epsilon^2 K)[K\alpha, -\epsilon\alpha]^T$ is

the solution of the lower triangular system

$$\begin{bmatrix} \epsilon^2 & 0 \\ \epsilon & K \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}. \tag{4.5}$$

Approximating (4.4) with (4.5) we obtain an easy to solve system whose solution is a good approximation to the solution of the original system.

Generalising this approach to higher dimensions we approximate (4.3) by

$$\begin{bmatrix} z^H(A-sI)z & 0 \\ W^H(A-sI)z & W^H(A-sI)W \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} z^H\hat{x}_1 \\ 0 \end{bmatrix}.$$

This block lower triangular system can be solved by forward substitution, that is, by solving

$$\begin{aligned} (z^HAz - s)p &= z^H\hat{x}_1, \\ W^H(A-sI)Wq &= -W^H(A-sI)zp, \end{aligned}$$

and the solution of the original problem is then $y = zp + Wq$.

Remarks

(i) $(z^HAz - s)$ is a scalar. Despite being ill-conditioned the first system is easy to solve.

(ii) The second system can be left multiplied by W to give

$$\mathcal{Q}(A-sI)Wq = -\mathcal{Q}(A-sI)zp.$$

which can be solved for $\tilde{q} = Wq$ using, for example, GMRES. In this way $\tilde{q}$ can be computed without explicitly computing W, since $\mathcal{Q} = I - \mathcal{P}$. The solution of the original system is then approximated by $y = zp + \tilde{q}$.

Using the above method for the solve in the Rayleigh Quotient Iteration (Algorithm

Algorithm 4.1: Split Rayleigh Quotient Iteration

Choose initial guess vector $\hat{x}^{(0)}$.
Choose initial vector z and let $\mathcal{Q} = I - zz^H$.
1. For $k = 1, 2, \ldots$ do
   a) Compute $\rho^{(k-1)} = \rho(\hat{x}^{(k-1)})$,
   b) i) Solve $(z^H A z - \rho^{(k-1)})\, \tilde{p}^{(k)} = z z^H \hat{x}^{(k-1)}$, and let $p^{(k)} = z\tilde{p}^{(k)}$,
      ii) Solve $\mathcal{Q}(A - \rho^{(k)} I)\mathcal{Q}\, q^{(k)} = -\mathcal{Q}(A - \rho^{(k)} I)p^{(k)}$,
      iii) Let $y^{(k)} = p^{(k)} + q^{(k)}$,
   c) Normalize, $\hat{x}^{(k)} = y^{(k)}/\|y^{(k)}\|_2$,
   d) Test for convergence of $\hat{x}^{(k)}$.

1.6) leads to the Split Rayleigh Quotient Iteration given in Algorithm 4.1.

4.3.2 Numerical experiments

We have applied the Split Rayleigh Quotient Iteration to two test problems which illustrate its performance. We first remark that this method is designed for the case when the shift s is very close to the desired eigenvalue and we only apply it in this case. This happens when the Rayleigh Quotient Iteration or Inverse Iteration are close to convergence.

In the following examples we apply Inverse Iteration with shift $s = 0$ and starting vector $[1, 1, \ldots, 1]^T$ until the Rayleigh Quotient of the current approximate eigenvector is close to the desired eigenvalue. At this point we switch to the Split Rayleigh Quotient Iteration.

We will measure the cost of applying the Split Rayleigh Quotient Iteration in *flops*, where a flop is one real floating point operation, for example an addition or multiplication. This differs from the convention in, for example, Golub and Van Loan [29, Ch. 3] but matches the convention used by Matlab (see [38]). We will also use the number of matrix vector multiplications (*mvs*) as a measure of cost.

Example 4.2 We use the Split Rayleigh Quotient Iteration, as described above, to compute the eigenvalue closest to zero of the matrix A =`diag([-4:0.05:-3,-0.05])`. This matrix has its spectrum in a cluster which is well away from the desired outlying eigenvalue -0.05.

The residuals from the Split Rayleigh Quotient Iteration (labelled 1) and from the Rayleigh Quotient Iteration (labelled 0) are plotted against number of flops required and the number of mvs required in Figures 4-2 (a) and (b) respectively. Both methods are implemented using GMRES to solve the linear systems which arise. We see that the Split RQI computes a solution with fewer mvs than the RQI—this is due to the reduction in the number of steps required by GMRES. This produces a corresponding reduction in flops.

Example 4.3 We use the Split Rayleigh Quotient Iteration, as described above, to compute the eigenvalue closest to zero of the tridiagonal matrix A, given by the 100×100 matrix `tridiag(1,-7,1)` augmented on the diagonal with the element 0.01. This matrix has most of its spectrum in a cluster which is well away from the desired outlying eigenvalue 0.01.

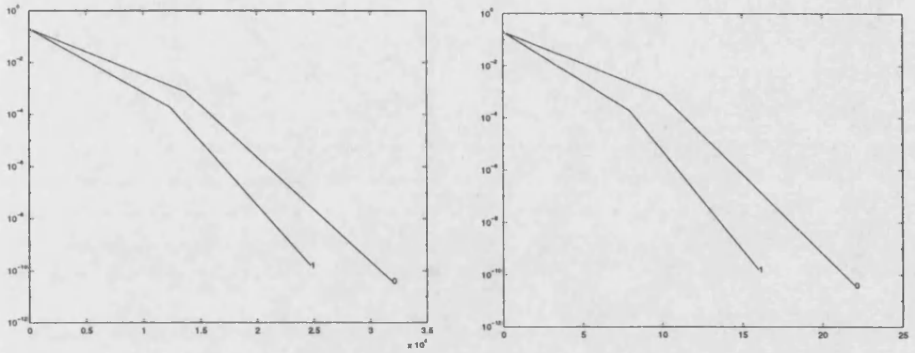
The residuals from the Split Rayleigh Quotient Iteration (labelled 1) and from the Rayleigh Quotient Iteration (labelled 0) are plotted against number of flops required and the number of mvs required in Figures 4-3 (a) and (b) respectively. We see the same qualitative behaviour that we saw in Example 4.2.

4.3.3 Larger Split Sizes

When the desired eigenvalue lies in a cluster of eigenvalues the convergence rate of GMRES may be improved by removing more than one eigenvalue. The generalisation of Algorithm 4.1 for splitsize m is straightfoward, one simply replaces z with $Z = [\hat{x}_1, \ldots, \hat{x}_m]$ where $\hat{x}_1, \ldots, \hat{x}_m$ are approximations to the eigenvectors corresponding to the eigenvalues that we wish to remove.

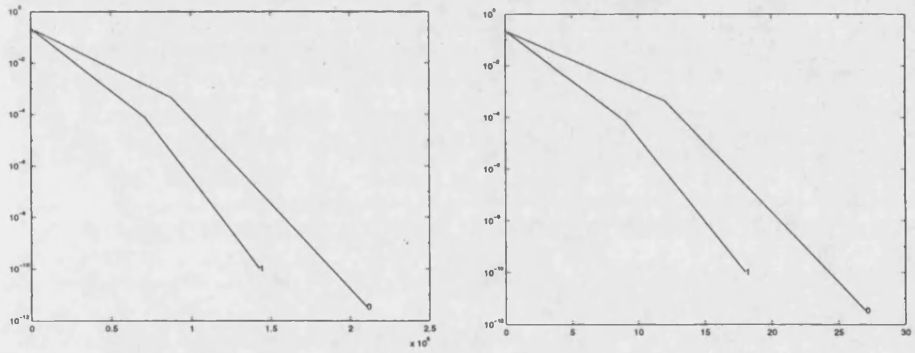
The generalised algorithm is applied with splitsize 2 in the following example.

Example 4.4 We use the Split Rayleigh Quotient Iteration, as described above, to compute the eigenvalue closest to zero of the tridiagonal matrix A, given by an 84×84 test matrix which has a dense cluster of eigenvalues at -10 and outlying eigenvalues at $\pm 2i$. This matrix is discussed in more detail in Example 3.4.



(a) (b)

Figure 4-2: Residual plotted against (a) no. flops and (b) no. mvs for Example 4.2



(a) (b)

Figure 4-3: Residual plotted against (a) no. flops and (b) no. mvs for Example 4.3

The residuals from the Split Rayleigh Quotient Iteration (labelled 2) and from the Rayleigh Quotient Iteration (labelled 0) are plotted against number of mvs required and the number of flops required in Figures 4-4 and 4-5 respectively. We see that although the Split RQI produces a satisfactory approximate eigenvalue more cheaply than the standard RQI the approximation obtained is not as good. In the split RQI we approximate the optimal solve in order to reduce cost—in this case the approximation reduces the quality of the eigenpair computed.

4.4 Summary

We have considered the use of iterative solvers to solve the linear systems which arise in Shift-Invert methods such as Inverse Iteration and the Rayleigh Quotient Iteration. The technique considered involves splitting the solve in such a way that we must solve

(i) with the restriction of $(A - sI)$ onto a space over which it is well conditioned. It is appropriate to use an iterative solver here, and the system obtained is of the form considered in Chapter 3 where we showed that such systems can be solved much more cheaply than the original system.

(ii) with the restriction of $(A - sI)$ onto a *small* space over which it is ill conditioned. This solve must be solved directly but the dimension of the system is small—direct solves are easily applied.

In Section 4.3 we showed how the above systems may be obtained in the Rayleigh Quotient Iteration. The result is a simple algorithm which implements the Rayleigh Quotient Iteration using GMRES on systems of the form (i). This algorithm computes eigenvalues more cheaply than the standard RQI using GMRES to solve with $(A - sI)$.

The algorithm obtained is applied to simple test problems for which it computes solutions more cheaply then the standard RQI when it is applied with GMRES.

The algorithm developed here is too simple to be considered for practical applications and is not robust. In addition, the approximation used in the algorithm becomes too great when we attempt to use split sizes greater than 1. However, the approach

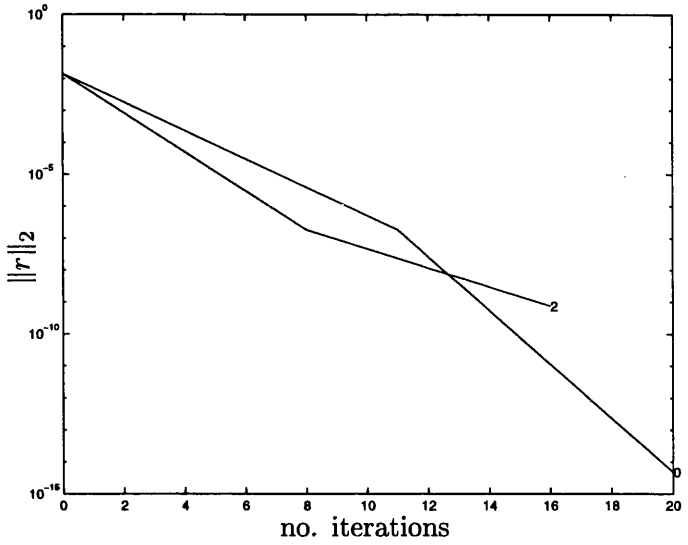


Figure 4-4: Residual norm against no. mvs for Example 4.4.

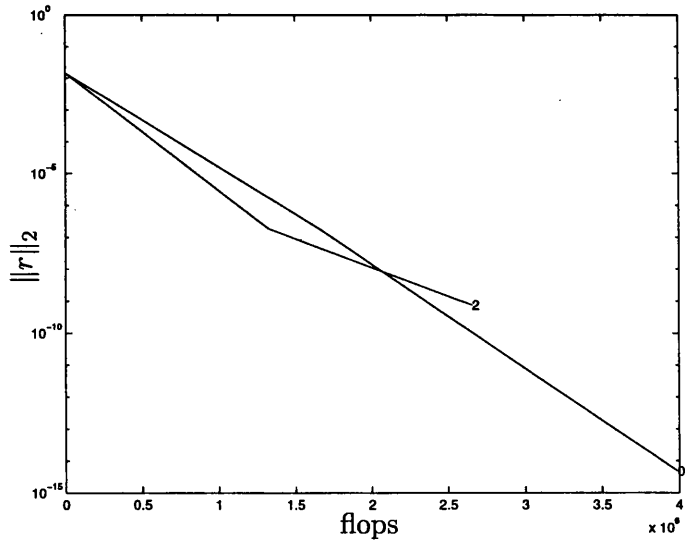


Figure 4-5: Residual norm against flops for Example 4.4.

illustrated in this algorithm is important and we return to the idea of *splitting* a solve in Shift-Invert methods in the following chapter.

Chapter 5

Splitting the Accelerated Rayleigh Quotient Iteration

5.1 Introduction

Is it possible to efficiently use iterative solvers to solve the near singular systems which arise when computing eigenvalues using methods based on the *Shift-Invert* transformation? In Chapter 3 we showed that iterative solvers can work efficiently on special systems derived from near singular systems. In the previous chapter we showed how we may obtain such systems in performing Inverse Iteration and we now show how we may obtain such systems when performing the Accelerated Rayleigh Quotient Iteration. This leads us to develop a new method for computing the eigenvalues of large, sparse matrices.

An extension of the approach developed in Chapter 4 is used to apply, using iterative solves, the Accelerated Rayleigh Quotient Iteration. This gives a new method which we call the Iterative Accelerated Rayleigh Quotient Iteration.

When carefully implemented the new method is cheaper to implement than the Accelerated Rayleigh Quotient Iteration but displays the same convergence rate. It is interesting to note that the Iterative ARQI generalises the Jacobi-Davidson method [9], [24], [64], [66].

An outline for this chapter is as follows. We begin, in Section 5.2, by briefly

reviewing the ideas of Chapter 4 and considering how they may be applied to the Accelerated Rayleigh Quotient Iteration. We introduce the projections that we will use to construct systems of the form described in Section 3.4.5. We then apply these projections to the linear system solved in the ARQI.

In Section 5.3 we show how the coupled systems obtained in Section 5.2 may be decoupled. There are two cases to be considered here:

(i) The general case: decoupling the systems requires us to make an approximation.

(ii) A special case: the decoupling used in (i) is exact.

We consider case (i) in Section 5.3, and consider case (ii) separately in Section 5.3.2.

The method for solving the coupled systems is the same for both cases. In Section 5.4 we develop the Iterative Accelerated Rayleigh Quotient Iteration Algorithm (IARQI—Algorithm 5.2) from the method described in Section 5.3 and discuss its implementation.

A convergence analysis for the IARQI is given in Section 5.5. The convergence analysis for case (ii) is based on the equivalence of IARQI with ARQI. The convergence analysis for case (i) is more complicated and uses results by Dembo and Eisenstat [17] on Inexact Newton methods.

In Section 5.6 we analyse the cost of implementing IARQI and discuss the importance of the results of Chapter 3 in estimating the overall implementation cost. Numerical results for the Iterative ARQI method are given in Section 5.7. In Section 5.8 we use the understanding developed in Chapter 3 and Section 5.6 of the implementation costs to discuss efficient ways of implementing the method.

5.2 Splitting the Accelerated RQI

In Section 4.2.3 we outlined a method which allows the use of iterative solvers to solve systems like (4.1)

$$(A - sI)y = \hat{x}.$$

Such systems arise in Shift-Invert methods for the eigenvalue problem. In Section 4.3 this method was developed for Inverse Iteration and the Rayleigh Quotient Iteration and an approximation to the desired eigenvector was used to *split* $\mathbb{C}^n$ into two spaces, the smaller space having dimension 1, and the larger space having dimension $(n-1)$. We replaced (4.1) with a solve on each of these subspaces—on the 1 dimensional space we use direct solves and on the $(n-1)$ dimensional space we use GMRES. This is a splitting with a split size of one.

In this section we expand on the technique of Section 4.3 by using split sizes greater than one. We also update the projections $\mathcal{P}$ and $\mathcal{Q}$ at each step of the iteration.

Suppose then that the split size m is greater than one. Using the outline in Section 4.2.3 we require, each time we solve (4.1), approximations $\hat{x}_1, \ldots, \hat{x}_m$ to the eigenvectors $x_1, \ldots, x_m$ of A. Moreover, we wish to update these approximations at each step of the iteration.

In Inverse Iteration and the Rayleigh Quotient Iteration we compute at each step only one approximate eigenvector—to compute more than one approximate eigenvector we require a more sophisticated algorithm. We consider a variant of the Rayleigh Quotient Iteration which computes a subspace rather than a single vector. From this subspace we may extract a number of approximate eigenpairs of A using the Rayleigh-Ritz procedure (see Chapter 1). The approximate eigenpairs computed in this way are called Ritz pairs. The resulting method is related to the Rayleigh Quotient Iteration in the same way that Arnoldi's method is related to the Power method. An implementation of this method is given in Algorithm 5.1. We call this method the Accelerated Rayleigh Quotient Iteration (Accelerated RQI or ARQI). The Accelerated RQI is a Rational Krylov method (see Ruhe [53, 52]).

In line (2a) of Algorithm 5.1 we solve at each step k the system

$$(A - \theta_1^{(k-1)} I) y^{(k)} = \hat{x}_1^{(k-1)}. \tag{5.1}$$

For now we will assume that each time we solve (5.1) we have computed the m approximate eigenvectors $\hat{x}_1^{(k)}, \ldots, \hat{x}_m^{(k)}$ at the previous step of the algorithm. This is

Algorithm 5.1: Accelerated Rayleigh Quotient Iteration

Choose initial guess vector $\hat{x}_1^{(0)}$.
Let $V_0 = [\hat{x}_1^{(0)}]$.
1. Compute $\theta_1^{(0)} = \rho(\hat{x}_1^{(0)})$,
2. For $k = 1, 2, \ldots$ do
   a) Solve $(A - \theta_1^{(k-1)} I) y^{(k)} = \hat{x}_1^{(k-1)}$,
   b) Let $V_k = \text{mgs}([V_{k-1}, y^{(k)}])$,
   c) Compute the Ritz Pair $(\hat{x}_1^{(k)}, \theta_1^{(k)})$ using the Rayleigh-Ritz procedure,
   d) Test for convergence.

the case when the dimension of the subspace $\mathcal{R}(V_k)$ is m or greater—then $k+1 \geq m$ approximate eigenvectors are computed by the Rayleigh-Ritz procedure at line (2c) of Algorithm 5.1. We will consider the problem when the dimension of the subspace $\mathcal{R}(V_k)$ is less than m in the next section.

Let us construct, as described in Section 4.2.3, the orthogonal projection $\mathcal{P} = ZZ^H$, where Z is an $n \times m$ orthonormal matrix with the same range as the matrix $[\hat{x}_1^{(k)}, \ldots, \hat{x}_m^{(k)}]$. Let us also construct the orthogonal projection $\mathcal{Q} = I - \mathcal{P}$. The projections $\mathcal{P}$ and $\mathcal{Q}$ define subspaces $\mathcal{F} = \mathcal{R}(\mathcal{P})$ and $\mathcal{G} = \mathcal{R}(\mathcal{Q})$ which satisfy

(i) $\mathbb{C}^n = \mathcal{G} \oplus \mathcal{F}$,

(ii) $\mathcal{G} \perp \mathcal{F}$.

The projections $\mathcal{P}$ and $\mathcal{Q}$ change at each step of the algorithm. However, for clarity we drop the superscripts and use the notation $\mathcal{P}$, $\mathcal{Q}$ in place of $\mathcal{P}^{(k)}$, $\mathcal{Q}^{(k)}$.

In this section we consider solving the system arising at a particular step k of Algorithm 5.1. For convenience we again drop the superscripts referring to the step k and so for the remainder of this section we use the terms $p, q, y, \hat{x}_i, \theta$ in place of $p^{(k)}, q^{(k)}, y^{(k)}, \hat{x}_i^{(k-1)}, \theta^{(k-1)}$.

In the following proposition we use the projections $\mathcal{P}$ and $\mathcal{Q}$ to split the system (5.1) over the spaces $\mathcal{F}$ and $\mathcal{G}$.

Proposition 5.1

Let $\mathcal{P}, \mathcal{Q}, \mathcal{F}$ and $\mathcal{G}$ be defined as above. Then the system (5.1)

$$(A - \theta_1 I)y = \hat{x}_1$$

is equivalent to the coupled systems

$$\begin{aligned} \mathcal{P}(A - \theta_1 I)p + \mathcal{P}(A - \theta_1 I)q &= \hat{x}_1, & (5.2) \\ \mathcal{Q}(A - \theta_1 I)p + \mathcal{Q}(A - \theta_1 I)q &= 0, & (5.3) \end{aligned}$$

where $p = \mathcal{P}y$ and $q = \mathcal{Q}y$.

Proof

We first apply the projections $\mathcal{P}$ and $\mathcal{Q}$ to (5.1). Then

$$\begin{aligned} \mathcal{P}(A - \theta_1 I)y &= \mathcal{P}\hat{x}_1, \\ \mathcal{Q}(A - \theta_1 I)y &= \mathcal{Q}\hat{x}_1. \end{aligned}$$

Now, writing $y = p + q$ where $p = \mathcal{P}y$ and $q = \mathcal{Q}y$ yields

$$\begin{aligned} \mathcal{P}(A - \theta_1 I)p + \mathcal{P}(A - \theta_1 I)q &= \mathcal{P}\hat{x}_1, \\ \mathcal{Q}(A - \theta_1 I)p + \mathcal{Q}(A - \theta_1 I)q &= \mathcal{Q}\hat{x}_1. \end{aligned}$$

Finally, we observe that $\mathcal{P}\hat{x}_1 = \hat{x}_1$ since $\hat{x}_1 \in \mathcal{F}$. Also $\mathcal{Q}\hat{x}_1 = 0$. □

5.3 Decoupling the systems

We now consider a technique for computing p and q in equations (5.2) and (5.3). We decouple these equations by observing that the term $\mathcal{Q}(A - \theta I)p$ in equation (5.3) is an approximation to a multiple of a known vector. In fact there is a special case where $\mathcal{Q}(A - \theta I)p$ is not just an approximation but is exactly a scalar multiple of

this known vector. With this observation a vector $\tilde{q}$ in the same direction as q may easily be computed. In fact, we show at the end of this section that it is not necessary to compute p in order to implement a split variant of Algorithm 5.1. However, for completeness we do explain how p may be recovered.

5.3.1 The general case

We begin with the general case and first show that it is easy to compute the direction of q. To see this, note that p is a linear combination of the m approximate eigenvectors $\hat{x}_1, \ldots, \hat{x}_m$. Thus there exist scalars $\alpha_1, \ldots, \alpha_m$ such that

$$p = \sum_{i=1}^{m} \alpha_i \hat{x}_i.$$

It is important to the following theory that the approximate eigenvectors $\hat{x}_1, \ldots, \hat{x}_m$ originate from a Rayleigh-Ritz process. Then with each $\hat{x}_i$ there is associated a Ritz value θ_i and a residual

$$r_i := A\hat{x}_i - \theta_i \hat{x}_i.$$

Each residual r_i is orthogonal to $\hat{x}_1, \ldots, \hat{x}_m$. It follows that $\mathcal{Q}r_i = r_i$ for $i = 1, \ldots, m$.

Combining these observations we have that

$$\begin{aligned}
(A - \theta_1 I)p &= (A - \theta_1 I) \sum_{i=1}^{m} \alpha_i \hat{x}_i \\
&= \sum_{i=1}^{m} \alpha_i (A - \theta_1 I)\hat{x}_i \\
&= \sum_{i=1}^{m} \left[A - \theta_i I + (\theta_i - \theta_1)I\right] \hat{x}_i \\
&= \sum_{i=1}^{m} \alpha_i \left[r_i + (\theta_i - \theta_1)\hat{x}_i\right].
\end{aligned}$$

Applying $\mathcal{Q}$ yields

$$\mathcal{Q}(A - \theta_1 I)p = \sum_{i=1}^{m} \alpha_i r_i. \tag{5.4}$$

Now suppose that $\hat{x}_1$ is a good approximation to x_1. Then $A\hat{x}_1 \approx \lambda_1 \hat{x}_1$ and so $(A - \theta_1 I)\hat{x}_1 \approx (\lambda_1 - \theta_1)\hat{x}_1$. Thus

$$\frac{1}{\lambda_1 - \theta_1}\hat{x}_1 \approx (A - \theta_1 I)^{-1}\hat{x}_1.$$

Suppose also that θ_1 is a good approximation to λ_1. Then $1/(\lambda_1 - \theta_1)$ is large and $y = (A - \theta_1 I)^{-1}\hat{x}_1$ will have a large component in $\hat{x}_1$. We expect that the $\hat{x}_1$ component of y is much larger than the $\hat{x}_2, \ldots, \hat{x}_m$ components of y. Thus $\alpha_1 \gg \alpha_2, \ldots, \alpha_m$ and it follows from (5.4) that

$$\mathcal{Q}(A - \theta_1 I)p \approx \alpha_1 r_1.$$

We then approximate equation (5.3) by

$$\mathcal{Q}(A - \theta_1 I)\tilde{q} = -\alpha_1 r_1. \tag{5.5}$$

The scalar α_1 depends on p, and so technically the systems (5.2) and (5.5) remain coupled. However, the approximation $\tilde{q}$ to q is coupled to p only by its magnitude.

5.3.2 Decoupling the systems without approximation

There is an important special case where there is *no* approximation, that is, that $\tilde{q}$ in equation (5.5) is a scalar multiple of q in equation (5.3). Thus there exists a scalar α such that

$$\mathcal{Q}(A - \theta_1 I)q = -\alpha r_1.$$

Recall from Chapter 2 the definition of the Krylov subspace. We showed in Chapter 1 that *all* Ritz pairs of a given matrix, computed from a particular Krylov subspace,

have residuals which lie in precisely one direction. We may now use the following results to show that we have precisely this situation—the residuals generated by Algorithm 5.1 lie in a single direction.

Theorem 5.2

At the kth step of Algorithm 5.1 the space spanned by the columns of the matrix V_k is a Krylov subspace.

Proof The proof of Theorem 5.2 is long, and we defer it until Section 5.10. □

Theorem 5.3

The residuals of the Ritz pairs computed in line (2c) at step k of Algorithm 5.1 are in the same direction.

Proof This is an immediate corollary of Theorem 5.2 and Lemma 1.5. □

These results mean that it is possible to compute exactly the direction of q by solving the system

$$\mathcal{Q}(A - \theta_1 I)\mathcal{Q}\ \tilde{q} = -r_1. \tag{5.6}$$

The vector $\tilde{q}$ lies in the same direction as the vector q in equations (5.2) and (5.3). This is shown in the following theorem, the proof of which is constructive.

Theorem 5.4

The solution $\tilde{q}$ of (5.6),

$$\mathcal{Q}(A - \theta_1 I)\mathcal{Q}\ \tilde{q} = -r_1.$$

is a multiple of the vector q in equations (5.2) and (5.3).

Proof

Recall equation (5.4),

$$\mathcal{Q}(A - \theta_1 I)p = \sum_{i=1}^{m} \alpha_i r_i,$$

where the scalars $\alpha_1, \ldots, \alpha_m$ arise as components of p in the approximate eigenvectors $\hat{x}_1, \ldots, \hat{x}_m$. We first observe that by Theorem 5.3 the residuals $r_1, \ldots, r_m$ of the approximate eigenvectors $\hat{x}_1, \ldots, \hat{x}_m$ lie in a single direction. Thus there exist scalars $\xi_1, \ldots, \xi_m$ such that

$$\xi_1 r_1 = \xi_2 r_2 = \cdots = \xi_m r_m.$$

For convenience we normalise so that $\xi_1 = 1$. Now equation (5.4) gives

$$\mathcal{Q}(A - \theta_1 I)p = \left(\sum_{i=1}^{m} \alpha_i / \xi_i \right) r_1.$$

Let $\alpha = \sum_{i=1}^{m} \alpha_i / \xi_i$. Then

$$\mathcal{Q}(A - \theta_1 I)p = \alpha r_1.$$

Substituting into (5.3) and rearranging gives

$$\mathcal{Q}(A - \theta_1 I)\tilde{q} = -\alpha r_1.$$

The result follows on observing that $\mathcal{Q}\tilde{q} = \tilde{q}$. □

Recall that in line (2b) of Algorithm 5.1 the subspace $\mathcal{R}(V_{k-1})$ is extended at each step with the direction of the vector y. We have shown that we may approximate y by $\tilde{y} = p + \alpha\tilde{q}$ for some scalar α, and we now observe that p is a linear combination of the current approximate eigenvectors and so is already contained within the subspace. In particular we have the following lemma.

Lemma 5.5

Let $\tilde{q}$ be computed as in Theorem 5.4. Then at line (2b) of Algorithm 5.1 we have

$$\text{mgs}([V_{k-1}, \tilde{y}]) = \text{mgs}([V_{k-1}, \tilde{q}]).$$

Proof

Recall that $\tilde{y} = p + \alpha\tilde{q}$ with $p \in \mathcal{R}(V_{k-1})$. Then

$$\text{mgs}([V_{k-1}, \tilde{y}]) = \text{mgs}([V_{k-1}, \tilde{q}]).$$

□

We have shown that if we solve (5.1) in Algorithm 5.1 using a split technique then we need only compute $\tilde{q}$. Substituting $\tilde{q}$ for q in equation (5.2) gives

$$\mathcal{P}(A - \theta_1 I)\tilde{p} + \mathcal{P}(A - \theta_1 I)\tilde{q} = \hat{x}_1, \tag{5.7}$$

where $\tilde{p} \in \mathcal{F}$ approximates p. In practise we need not compute $\tilde{p}$, but for completeness we show how this may be done in Section 5.11 at the end of this chapter.

5.4 Implementation

We now discuss the implementation of Accelerated RQI using split solves. We recall here that the objective in using split solves is to replace the potentially difficult or expensive solve (5.1)

$$(A - \theta_1 I)y = \hat{x}_1$$

of line (2a) in Algorithm 5.1 with a computationally cheaper split solve. In the previous section we proposed a way of splitting (5.1) which produces equivalent results. In particular we showed that it is sufficient to solve (5.6)

$$\mathcal{Q}(A - \theta_1 I)\mathcal{Q}\ \tilde{q} = -r_1$$

where $\mathcal{Q}$ is some appropriately chosen projection. In this way we may implement a split, or iterative, variant of ARQI with only a change to line (2a) of Algorithm 5.1.

In order to construct the projection $\mathcal{Q}$ we require at least m approximate eigenvectors—these are available when the dimension of the subspace $\mathcal{R}(V_k)$ is m or greater, that is, at step $m - 1$ of the algorithm. This problem was outlined in the

Algorithm 5.2: Iterative Accelerated Rayleigh Quotient Iteration

Choose initial guess subspace with basis matrix V_l.

1. Compute the Ritz pairs $(\hat{x}_1^{(l)}, \theta_1^{(l)}), \ldots, (\hat{x}_l^{(l)}, \theta_l^{(l)})$, and the residual r_1.
2. For $k = l+1, l+2, \ldots$ do
   a) i) Compute $\mathcal{Q}$ from $\hat{x}_1^{(k-1)}, \ldots, \hat{x}_m^{(k-1)}$.
      ii) Solve $\mathcal{Q}(A - \theta_1^{(k-1)} I)\mathcal{Q}\ \tilde{q} = -r_1$,
   b) Let $V_k = \text{mgs}([V_{k-1}, \tilde{q}])$,
   c) Compute the Ritz pairs $(\hat{x}_1^{(k)}, \theta_1^{(k)}), \ldots, (\hat{x}_k^{(k)}, \theta_k^{(k)})$, and the residual r_1,
   d) Test for convergence.

previous section and we return to it now.

To solve the problem, that is, to make available at line (2a) of Algorithm 5.1 at least m approximate eigenvectors, we start the algorithm with not one initial vector but with $l \geq m$ initial vectors. We replace the initial guess vector $\hat{x}_1^{(0)}$ of Algorithm 5.1 by an initial guess *subspace* with dimension m or greater. In line (1) of the algorithm we then apply the Rayleigh-Ritz procedure to compute the required approximate eigenvectors.

The choice of initial guess subspace is important to the convergence characteristics of the algorithm. In principle, any technique may be applied to generate the initial guess subspace, for example $m-1$ steps of Arnoldi's method. Such techniques lead the new method to produce a subspace which is *not* a Krylov subspace, and the solution $\tilde{q}$ of (5.6) merely approximates some scalar multiple of $q = \mathcal{Q}y$. We will discuss the effects of this approximation on the convergence rate in more detail in Section 5.5. If the guess subspace is computed using a Shift-Invert method, for example using Algorithm 5.1, then Theorems 5.2, 5.3 and 5.4 show that the solve (5.6)

$$\mathcal{Q}(A - \theta_1 I)\mathcal{Q}\ \tilde{q} = -r_1$$

may replace (5.1), and that there is no approximation involved.

This method, which we call the *Iterative Accelerated Rayleigh Quotient Iteration* (IARQI), is implemented in Algorithm 5.2.

Remarks

(i) The solve (5.6) is intended to be performed using a Krylov solver such as GMRES. Such solvers require only the action of the matrix, in this case the action of $\mathcal{Q}(A - \theta_1 I)\mathcal{Q}$. Thus we need not explicitly form an $n \times n$ matrix representing $\mathcal{Q}$—the action of $\mathcal{Q}$ on a given vector is obtained each time it is required by performing m inner products.

(ii) The projection $\mathcal{Q}$ is given, as described in Section 4.2, by $\mathcal{Q} = I - ZZ^H$, where $\mathcal{R}(Z) = \langle \hat{x}_1, \ldots, \hat{x}_m \rangle$ and Z is an orthonormal matrix. Note that if the split size m is one then $\mathcal{Q} = I - \hat{x}_1 \hat{x}_1^H$ and the solve at line (2aii) of Algorithm 5.2 becomes

$$(I - \hat{x}_1 \hat{x}_1^H)(A - \theta_1 I)(I - \hat{x}_1 \hat{x}_1^H)\tilde{q} = -r_1. \tag{5.8}$$

When $m = 1$ IARQI reduces to the Jacobi-Davidson method.

(iii) The link between IARQI and the Jacobi-Davidson method allows the exploitation of a number of techniques developed for the solve (5.8) in Jacobi-Davidson.

One such technique developed for Jacobi-Davidson in Booten and Van der Vorst [6, 7, 18] is to solve (5.8) by reformulating it as the $(n+1) \times (n+1)$ bordered system

$$\begin{bmatrix} A - \theta_1 I & \hat{x}_1 \\ \hat{x}_1^H & 0 \end{bmatrix} \begin{bmatrix} \tilde{q} \\ \epsilon \end{bmatrix} = \begin{bmatrix} -r_1 \\ 0 \end{bmatrix}.$$

In the same way one can reformulate (5.6) as the $(n+m) \times (n+m)$ block system

$$\begin{bmatrix} A - \theta_1 I & Z \\ Z^H & 0 \end{bmatrix} \begin{bmatrix} \tilde{q} \\ \epsilon \end{bmatrix} = \begin{bmatrix} -r_1 \\ 0 \end{bmatrix}.$$

This reformulation has the same advantages and disadvantages as in the Jacobi-Davidson case which is discussed more fully in Chapter 2. In this chapter we will not consider the block form of (5.6).

5.5 Convergence analysis

We now present an analysis of the convergence rate of Algorithm 5.2. In doing this we consider the two natural cases:

(i) The general case, where we make no special assumptions about the initial guess subspace,

(ii) The special case where the initial guess subspace is produced with a shift-invert method.

When the split size m is greater than one the convergence rate of Algorithm 5.2 is different in these cases. We present the analysis for case (i) first, but the convergence analysis for case (ii) is much simpler than for case (i).

In both cases we assume that GMRES or an alternative Krylov solver is used, and that all solves are performed to high accuracy. We discuss the case where solves are performed inexactly separately at the end of this section.

5.5.1 Case (i) General initial guess subspace

Consider the case when we have a general initial guess subspace. Such subspaces might arise from Arnoldi's method or may be constructed from some previously computed approximate eigenvectors. In general the subspace computed by Algorithm 5.2 is *not* a Krylov subspace and the simpler convergence analysis that we will use in case (ii) does not apply.

We analyse the convergence of Algorithm 5.2 in this case by showing that at each step of the algorithm we are performing a step of an inexact Newton method. By analysing 5.2 as an accelerated inexact Newton method we show that the method has superlinear convergence.

Newton's method for the eigenvalue problem Consider the eigenvalue problem $Ax = \lambda x$, $\|x\|_2 = 1$. It is convenient to introduce the function $F : \mathbb{C}^{n+1} \to \mathbb{C}^{n+1}$

defined by

$$F(x,\lambda) = \begin{bmatrix} Ax - \lambda x \\ -\frac{1}{2}x^H x + \frac{1}{2} \end{bmatrix}. \tag{5.9}$$

Then (x,λ) satisfy $Ax = \lambda x$, $\|x\|_2 = 1$ if and only if $F(x,\lambda) = 0$.

Suppose that $(\hat{x},\theta)$ is an approximate solution of $F(x,\lambda) = 0$, and that θ is the Rayleigh Quotient of $\hat{x}$. To improve the approximate solution using Newton's method we solve the Newton system

$$\begin{bmatrix} A - \theta I & -\hat{x} \\ -\hat{x}^H & 0 \end{bmatrix} \begin{bmatrix} z \\ \epsilon \end{bmatrix} = - \begin{bmatrix} A\hat{x} - \theta\hat{x} \\ 0 \end{bmatrix}$$

and compute the new approximate solution $(\hat{x} + z, \theta + \epsilon)$.

Suppose that Z is an $n \times m$ full rank, orthonormal matrix with $\hat{x} \in \mathcal{R}(Z)$. Let W be an $n \times (n-m)$ orthonormal, full rank matrix whose columns are orthogonal to those of Z. Then there exist $\psi \in \mathbb{C}^m$ and $\tau \in \mathbb{C}^{n-m}$ such that

$$\begin{bmatrix} z \\ \epsilon \end{bmatrix} = \begin{bmatrix} Z & W & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \psi \\ \tau \\ \epsilon \end{bmatrix}.$$

The Newton system above may be written

$$\begin{bmatrix} Z & W & 0 \\ 0 & 0 & 1 \end{bmatrix}^H \begin{bmatrix} A - \theta I & -\hat{x} \\ -\hat{x}^H & 0 \end{bmatrix} \begin{bmatrix} z \\ \epsilon \end{bmatrix} = - \begin{bmatrix} Z & W & 0 \\ 0 & 0 & 1 \end{bmatrix}^H \begin{bmatrix} r \\ 0 \end{bmatrix},$$

and, multiplying out and noting that since θ is the Rayleigh Quotient of $\hat{x}$,

$$\begin{bmatrix} Z^H(A-\theta I)Z & Z^H(A-\theta I)W & -Z^H\hat{x} \\ W^H(A-\theta I)Z & W^H(A-\theta I)W & 0 \\ -\hat{x}^H Z^H & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \tau \\ \epsilon \end{bmatrix} = - \begin{bmatrix} 0 \\ W^H r \\ 0 \end{bmatrix}. \tag{5.10}$$

Solving this system is equivalent to solving the coupled systems (5.2)/(5.3), where

$\psi = Z^H(p - \hat{x})$ and $\tau = W^H q$.

The decoupled systems (5.7) and (5.6) are equivalent to the system

$$\begin{bmatrix} Z^H(A-\theta I)Z & Z^H(A-\theta I)W & -Z^H\hat{x} \\ 0 & W^H(A-\theta I)W & 0 \\ -\hat{x}^H Z^H & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\psi} \\ \hat{\tau} \\ \epsilon \end{bmatrix} = - \begin{bmatrix} 0 \\ W^H r \\ 0 \end{bmatrix}, \qquad (5.11)$$

where $\hat{\psi} = Z^H(\tilde{p} - \hat{x})$ and $\hat{\tau} = W^H\tilde{q}$. Equations (5.10) and (5.11) differ only in the (2,1) term of the Jacobian matrix. Using (5.11) in place of (5.10) leads to an inexact Newton method.

Dembo, Eisenstat, and Steihaug [17] give convergence results for the following inexact Newton method for the problem $F(x) = 0$ (which has solution $x^\star$):

For $k = 0, 1, 2, \ldots$ until convergence do

- Find $s^{(k)}$ which satisfies

$$F_x(x^{(k)})s^{(k)} = -F(x^{(k)}) + e^{(k)}$$

  where $\|e^{(k)}\|_2 / \|F(x^{(k)})\|_2 \leq \eta_k$,

- let $x^{(k+1)} = x^{(k)} + s^{(k)}$.

The convergence rate of the inexact Newton method is characterised by the *forcing sequence* $\{\eta_k\}$. Dembo et al. prove the following result.

Theorem 5.6 (Dembo et al. [17, Corollary 3.5])

Assume that the inexact Newton iterates $\{x^{(k)}\}$ *converge to* $x^\star$. *Then* $x^{(k)} \to x^\star$ *superlinearly if* $\lim_{k\to\infty} \eta_k = 0$

By showing that the inexact Newton method given by (5.11) for the problem (5.9) satisfies the assumptions of Theorem 5.6 we will prove that the approximate eigenpair computed by this inexact Newton method converges superlinearly to an eigenpair of A.

Theorem 5.7

Let the sequence $[(\hat{x}^{(k)})^T, \theta^{(k)}]^T$ *be generated by inexact Newton for (5.9) implemented by solving the system (5.11) exactly. Assume that*

(i) $[(\hat{x}^{(k)})^T, \theta^{(k)}]^T$ *converges to an eigenpair of* A,

(ii) the columns of the $n \times m$ *orthonormal matrix* $Z^{(k)} = [z^{(k)}{}_1, \ldots, z^{(k)}{}_m]$ *have residuals (for the eigenvalue problem)* $r^{(k)}{}_1, \ldots, r^{(k)}{}_m$ *satisfying*

$$\|r_i^{(k)}\|_2 \leq \kappa_{ik}(Ax^{(k)} - \theta^{(k)}x^{(k)}),$$

for some scalars κ_{ik}. *Without loss of generality we assume* $z_1^{(k)} = x^{(k)}$.

Then the sequence $\{[(\hat{x}^{(k)})^T, \theta^{(k)}]^T\}$ *converges superlinearly to an eigenpair of* A.

Proof For clarity we will drop the superscripts denoting the step k of the inexact Newton method.

Let $r = Ax - \theta x$ and write $s = [z^T, \epsilon]^T$ where $z \in \mathbb{C}^n$ and $\epsilon \in \mathbb{C}$. Let $\psi = Z^H z \in \mathbb{C}^m$ and $\tau = W^H z \in \mathbb{C}^{n-m}$. The error term e satisfies

$$e = F_{x,\lambda}(x, \theta)s + F(x, \theta),$$

and it is straightforward to show that $\|e\|_2 \leq \|W^H(A - \theta I)Z\psi\|_2$. Now

$$\begin{aligned}
\|W^H(A - \theta I)Z\psi\|_2 &= \|\sum_{i=1}^{m} W^H r_i \psi_i\|_2 \\
&\leq \sum_{i=1}^{m} \|W^H r_i\|_2 |\psi_i| \\
&\leq \sum_{i=1}^{m} \kappa_{ik}\|r_1\|_2 |\psi_i| \\
&\leq \|r_1\|_2 \sum_{i=1}^{m} \kappa_{ik}|\psi_i|.
\end{aligned}$$

Since $[(\hat{x}^{(k)})^T, \theta^{(k)}]^T$ converges to an eigenpair of A we have that $z^{(k)} \to 0$ as $k \to \infty$. Thus $\psi^{(k)} = Z^{(k)H} z^{(k)} \to 0$. Observing that $\|F(x, \theta)\|_2 = \|[r^T, 0]^T\|_2 = \|r_1\|_2$ we have

that

$$\frac{\|e^{(k)}\|_2}{\|F(x^{(k)}, \theta^{(k)})\|_2} =: \eta_k \to 0 \quad \text{as } k \to \infty.$$

The result follows by Theorem 5.6. □

5.5.2 Case (ii) Initial guess subspace generated by a Shift-Invert method

Consider the case where the initial guess subspace is generated by $l \geq m$ steps of a shift-invert method. Such a space is spanned by the vectors $\psi_1, \ldots, \psi_{l+1}$ where ψ_1 is the starting vector, and the remaining ψ_j satisfy

$$(A - s_j I)\psi_j = \sum_{i=1}^{j-1} \alpha_{ij}\psi_i,$$

where s_j is a shift and $\sum_{i=1}^{j-1} \alpha_{ij}\psi_i$ is a linear combination of the previously computed vectors. With such an initial guess subspace the subspace generated by IARQI is a Krylov subspace.

Theorem 5.8 *Let V_l be an orthonormal matrix whose columns span $\langle \psi_1, \ldots, \psi_l \rangle$ as defined above. Then the subspace generated at step k of Algorithm 5.2, using $\langle \psi_1, \ldots, \psi_l \rangle$ as an initial guess space, is a Krylov subspace.*

Proof

The extension of the initial guess subspace using Algorithm 5.2 is a continuation of a general shift-invert method. The result follows by Theorem 5.11 which is deferred until the end of this chapter. □

This result shows that the residuals of *any* two approximate eigenvectors computed by IARQI are in the same direction. Consequently the correction $\tilde{q}$ computed by IARQI is in the same direction as the true correction q, and the Accelerated Rayleigh Quotient Iteration and IARQI are equivalent. It follows that the two methods share the same convergence properties.

Theorem 5.9

Algorithm 5.2 with an initial guess subspace generated by a shift-invert method converges cubically if A is symmetric and quadratically if A is nonsymmetric.

Proof

From Theorem 1.5, Theorem 5.4, and Lemma 5.12, we have that IARQI and ARQI are equivalent at each step. ARQI converges cubically if A is symmetric and quadratically if A is nonsymmetric. □

5.5.3 Inexact solves

When solves are performed inexactly the situation is very similar to case (i) above. The IARQI can be thought of as an accelerated inexact Newton method, with the forcing sequence determined not only by the choice of projection $\mathcal{Q}$ but also by the accuracy of the approximate solves.

5.6 Cost analysis

Example 5.1 In this example A is a real, diagonal, 100 × 100 matrix. A has 99 eigenvalues between 10 and 11, and one eigenvalue at 1e-4. The vector z is a normalised approximation to the eigenvector of A which has corresponding eigenvalue 1e-4. The vector b is given by $b = (I - zz^H)\texttt{ones(100,1)}$. We compare the convergence rate of GMRES for the systems (i) $Ax = b$ and (ii) $(I - zz^H)A(I - zz^H)x = b$. The second system is typical of the systems which arise in Algorithm 5.2. The first system provides a comparison.

Figure 5-1 shows the convergence history of GMRES for systems (i) and (ii). GMRES converges in 10 steps for system (i) and in 5 steps for system (ii).

Figure 5-2 (a) shows the total number of floating point operations used to compute an approximate solution $x^{(k)}$ at step k of GMRES. The curves in Figure 5-2 (a) are nonlinear since the orthonormalisation costs in GMRES increase with the size of the subspace. We see that at each step, the cost of computing an $x^{(k)}$ is greater for system

(ii) than for system (i). This is because multiplication by $(I - zz^H)A(I - zz^H)$ requires more floating point operations than multiplication by A. The difference between these two curves is shown in figure 5-2 (b). This curve is linear and represents the difference in cost of multiplying by $(I - zz^H)A(I - zz^H)$ and multiplying by A.

In order to gain benefit from using the split solves we require that the extra cost of applying $\mathcal{Q}A\mathcal{Q}$ to a vector be offset by a reduction in the number of steps of GMRES require to compute a satisfactory solution.

As before, we will measure the cost of matrix multiplication in *flops*, where a flop is one real floating point operation, for example an addition or multiplication.

Matrix Multiplication Costs Recall that A is an $n \times n$ real or complex matrix. Suppose that A has bandwidth l; then the cost of multiplying a vector by A is $\mathcal{O}(2ln)$ flops if the result is real, and $\mathcal{O}(8ln)$ flops if the result is complex.

Projected Matrix Multiplication Costs We consider here the projected matrix $\mathcal{Q}A\mathcal{Q}$ where $\mathcal{Q}$ is given by $(I - ZZ^H)$ for some orthonormal $n \times m$ matrix $Z = [z_1, \ldots, z_m]$. Given the vector x we compute $\mathcal{Q}x$ by:
For $i = 1, 2, \ldots, m$,

- compute $y = z_i^H x$,
- compute $w = z_i y$,
- update x as $x - w$.

In real arithmetic the three inner steps cost $\mathcal{O}(2n)$, $\mathcal{O}(n)$ and $\mathcal{O}(n)$ flops respectively. Thus the total cost of applying $\mathcal{Q}$ to a vector is $\mathcal{O}(4mn)$. The cost of multiplying a vector by the projected matrix is thus $\mathcal{O}(8mn + 2ln)$.

In complex arithmetic the three inner steps cost $\mathcal{O}(8n)$, $\mathcal{O}(6n)$, and $\mathcal{O}(2n)$ flops respectively. Here the total cost of applying $\mathcal{Q}$ to a vector is $\mathcal{O}(16mn)$. The total cost of multiplying a vector by the projected matrix is thus $\mathcal{O}(32mn + 8ln)$

Remarks

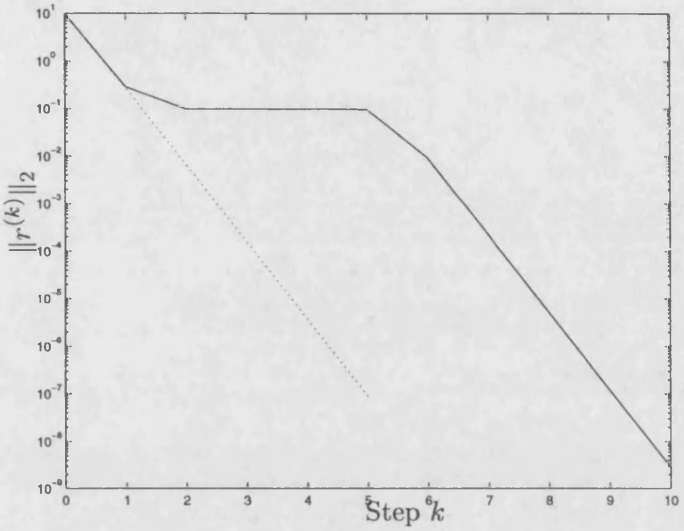
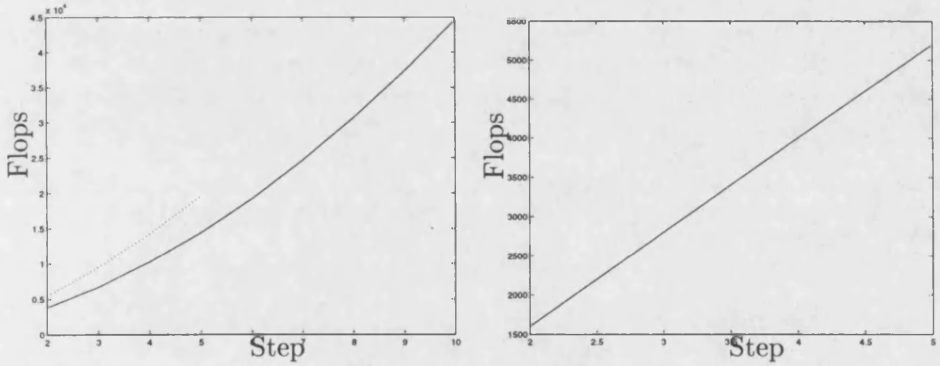


Figure 5-1: Convergence history of GMRES for system (i) solid line and system (ii) dotted line in Example 5.1.



(a) (b)

Figure 5-2: (a) Flop counts for system (i) solid line and system (ii) dashed line. (b) Flop count for system (i) subtracted from that for system (ii) in Example 5.1.

(i) At step k of GMRES (Algorithm 3.1) for $\mathcal{Q}A\mathcal{Q}x = b$ we multiply the vector $v^{(k)}$ by $\mathcal{Q}A\mathcal{Q}$. If the initial vector for GMRES is in $\mathcal{G}$ then $v^{(k)}$ has the property that $\mathcal{Q}v^{(k)} = v^{(k)}$. Thus $\mathcal{Q}A\mathcal{Q}v^{(k)} = \mathcal{Q}Av^{(k)}$ and in practise only *one* application of $\mathcal{Q}$ is required. This reduces the cost of multiplication by $\mathcal{Q}A\mathcal{Q}$ to $\mathcal{O}(4mn + 2ln)$ flops if the result is real and $\mathcal{O}(16mn + 8ln)$ flops if the result is complex.

(ii) The Matlab implementation of GMRES computes the true residual $b - \mathcal{Q}A\mathcal{Q}x^{(k)}$ of the approximate solution $x^{(k)}$ at each step k. Our algorithm at this point applies $\mathcal{Q}$ both times.

We now compare the total costs of applying k steps of GMRES for

(i) $Ax = b$,

(ii) $\mathcal{Q}A\mathcal{Q}x = b$.

Suppose that the cost of orthonormalising an n-vector against another vector is c_o flops. Suppose that the cost of multiplying by the matrix A is c_m flops.

At step k of a simple GMRES algorithm we perform

- one matrix vector multiplication with the iteration matrix,
- k orthonormalisations.

The matrix vector multiplication for system (ii) requires a further m orthonormalisations which cost mc_o flops. It follows that the total cost at step k of GMRES is

$$\begin{aligned} & \sum_{j=1}^{k} [mc_o + jc_o + c_m] \\ = \; & kmc_o + kc_m + \frac{k}{2}(k+1)c_o. \end{aligned}$$

If A is banded with bandwidth l then $c_m = \mathcal{O}(2ln)$. We also have $c_o = \mathcal{O}(4n)$. It is clear that as the cost of multiplication by A increases the cost of applying $\mathcal{Q}$ becomes less significant.

This analysis allows us to estimate the reduction in steps used which is required to make system (ii) attractive in place of system (i).

5.7 Numerical experiments

We have applied the IARQI (Algorithm 5.2) to a number of test problems. We begin by demonstrating particular behaviour of the method for small test problems with special eigenvalue distributions. We then present numerical results for matrices arising from practical problems.

Our examples demonstrate the following behaviour:

1. For problems with k outlying eigenvalues

   (a) split sizes up to k can reduce the total number of matrix vector multiplications required, with a corresponding reduction in total cost. The greatest reduction in mvs occurs for split size k.

   (b) split sizes greater than k do not reduce further the number of matrix vector multiplications required but do increase the cost.

2. When the outlying eigenvalues are not well separated from the rest of the spectrum the reduction in matrix vector multiplications obtained by increasing the split size is small. There is no significant reduction in cost.

3. We may generate initial guess subspaces using a shift-invert method or some other method. When a general method is used and the split size is greater than 1 the residuals obtained at each step are larger than those obtained using the accelerated Rayleigh Quotient Iteration (Algorithm 5.1). However, when we take into account the solutions produced from these types of initial space, together with the cost required to compute them, we usually see that IARQI is better.

In this section we use the following conventions:

- By IARQI with split sizes $m = 1, 2, \ldots$ we mean Algorithm 5.2 implemented with split size m.
- By IARQI with split size $m = 0$ we mean the Accelerated Rayleigh Quotient Iteration, Algorithm 5.1. This algorithm uses no splitting.

We begin with some examples which demonstrate the potential gains in removing eigenvalues.

Example 5.2 We use IARQI to compute the eigenvalue closest to zero of the matrix A =`diag([-4:0.05:-3,-0.05,-0.1])`. This matrix has its spectrum in a cluster which is well away from the desired eigenvalue, but has the pair of outlying eigenvalues -0.1 and -0.05. Our initial guess space is the 4 dimensional Krylov subspace generated with the matrix $(A - I)^{-1}$ and the initial vector $[1, 1, \ldots, 1]^T$.

1. Figure 5-3 (page 155) shows the residual norm against the number of mvs used for split sizes 2, 1, and 0 (ARQI) for this problem. We see that

   (i) the residual norms are at each step very similar,

   (ii) IARQI with split sizes 2 and 1 compute a solution respectively in 57 percent and 71 percent of the mvs required to compute a solution using split size 0.

   Figure 5-4 shows the residual norm against the number of flops used for split sizes 2, 1, and 0 (ARQI) for this problem. The cost of an mv increases with the split size. Since A is diagonal the cost of applying the projection Q is high compared with the cost of multiplying by A, but it is still significantly cheaper to compute a solution using IARQI with $m = 2$ than to use ARQI.

2. Figures 5-5 and 5-6 (page 156) show the residual norm against number of mvs and flops respectively for IARQI with split sizes 3, 2, 1 and 0. When a split size of 2 is used all of the outlying eigenvalues are projected out. When a split size of 3 is used an additional eigenvalue is projected out. This is not an outlying eigenvalue and Figure 5-5 shows that its removal does not reduce the number of mvs used. It is more expensive to implement IARQI with $m = 3$ than with $m = 2$. This can be seen in Figure 5-6.

Example 5.3 We use IARQI to compute the eigenvalue closest to zero of the tridiagonal matrix A, given by the 100×100 matrix `tridiag(1,-7,1)` augmented on the diagonal with the block `diag([0.01,0.02])`. This matrix has most of its spectrum in

a cluster which is well away from the desired eigenvalue, but has the pair of outlying eigenvalues 0.01 and 0.02. Our initial guess space is the 4 dimensional Krylov subspace generated with the matrix $(A-I)^{-1}$ and the initial vector $[1,1,\ldots,1]^T$.

1. Figure 5-7 (page 157) shows the residual norm against the number of mvs used for split sizes 2, 1, and 0. We see that

   (i) the residuals at each step are very similar,

   (ii) IARQI with split sizes 2 and 1 compute a solution respectively in 51 and 74 percent of the mvs required to compute a solution using split size 0.

   Figure 5-8 shows the residual norm against the number of flops used for split sizes 2, 1, and 0 for this problem. We see that IARQI with split sizes 2 and 1 compute a solution respectively in 60 and 77 percent of the flops required to compute a solution using split size 0.

2. Figures 5-9 and 5-10 (page 158) show the residual norm against number of mvs and flops respectively for IARQI with split sizes 3, 2, 1 and 0. We observe the same behaviour as in Example 5.2 part 2: increasing the split size from 2 to 3 increases the cost, and does not reduce the number of mvs required.

Example 5.4 Recall the matrix in Example 5.3. We now use IARQI to compute the eigenvalue closest to zero of the matrix derived from A by removing the last row and column, so removing one of the small eigenvalues. This matrix has most of its spectrum well separated from the desired eigenvalue, but has the single outlying eigenvalue 0.01. Our initial guess space is the 3 dimensional Krylov subspace generated with the matrix $(A-I)^{-1}$ and the initial vector $[1,1,\ldots,1]^T$.

We see here that the optimum split size is 1. With split size 1 we compute a solution in approximately 60 percent of the mvs and 65 percent of the flops required with split size 0. This is illustrated in Figures 5-11 and 5-12 (page 159).

We now give some examples which illustrate how the performance of IARQI depends on the separation of the desired eigenvalues from the remainder of the spectrum. We

begin with two examples where the separation is moderate and end with an example where the separation is small.

Example 5.5 We use IARQI to compute the closest eigenvalue to zero of the matrix obtained from the matrix in Example 5.2 by shifting the eigenvalues in the interval $[-4,-3]$ so that they are now in the interval $[-2,-1]$. Our initial guess space is the 4 dimensional Krylov subspace generated with the matrix $(A-I)^{-1}$ and the initial vector $[1,1,\ldots,1]^T$.

The residual norm is plotted against the number of mvs and the number of flops in Figures 5-13 and 5-14 (page 160) respectively. Comparing these with Figures 5-3 and 5-4 from Example 5.2, we see that in all cases the work required has increased. This is due to the closeness of the eigenvalues of the matrix to the origin. We see that the relative gain in using larger split sizes is decreased in comparison with Example 5.2.

Repeating this experiment with the spectrum again altered so that the eigenvalues which originally lay in the interval $[-4,-3]$ now lie in the interval $[-1.2,-0.2]$ shows similar behaviour. This is illustrated in Figures 5-15 and 5-16.

Example 5.6 We use IARQI to compute the closest eigenvalue to zero of the matrix obtained from the matrix in Example 5.3 by changing the diagonal entries which were -7 to -3. This moves the non-outlying eigenvalues of the matrix closer to the origin. Our initial guess space is the 4 dimensional Krylov subspace generated with the matrix $(A-I)^{-1}$ and the initial vector $[1,1,\ldots,1]^T$.

The residual norm is plotted against the number of mvs in Figure 5-17, and the number of flops in Figure 5-18 (page 162). We see, as in Example 5.5, a reduction in the relative gain obtained by increasing the split size and an overall increase in cost. The number of mvs required with split size 0 shows a very large increase. This is where GMRES has difficulties solving the near singular system in ARQI.

In practise it is difficult to compare the results obtained from IARQI with different initial guess subspaces because different subspaces will not contain approximate eigenvectors of the same quality. The following example illustrates that with an initial guess

space produced by Arnoldi's method on A, increasing the split size from 1 to 2 reduces the overall cost but also increases the residual norm.

Example 5.7 We use the matrix from Example 5.3 which has most of its spectrum in a cluster which is well away from the desired eigenvalue, but has the pair of outlying eigenvalues 0.01 and 0.02. Our initial guess space is the 6 dimensional Krylov subspace generated with the matrix A and the initial vector $[1, 1, \ldots, 1]^T$.

Figures 5-19 and 5-20 (page 163) show the residual norm plotted against number of mvs and flops respectively. We see that, although IARQI with $m = 2$ is cheaper than IARQI with $m = 1$ (which is in turn cheaper than the Accelerated RQI), the residual norm increases slightly with m.

We now present some experiments on matrices arising from practical problems. We consider the following matrices

lop163 A 163×163 real nonsymmetric matrix arising from Markov modelling techniques. This matrix is in the STOCH set of the NEP collection (see [3]).

cavity01 A 317×317 real nonsymmetric matrix arising in driven cavity problems. This matrix is in the DRIVCAV_OLD set of the SPARSKIT collection (see [55]).

fidap001 A 216×216 real nonsymmetric matrix arising in fluid dynamics. This matrix is in the FIDAP set of the SPARSKIT collection (see [55]).

gre185 A 185×185 real nonsymmetric matrix arising in the computer simulations. This matrix is in the GRENOBLE set of the Harwell–Boeing collection (see [20]).

Example 5.8 The spectrum of the matrix **lop163** is shown in Figure 5-21 (page 164). We see that the spectrum of the matrix is concentrated in a cluster at about 0.8, but with a line of real eigenvalues extending to approximately -0.8.

We apply IARQI with split sizes $0, 1, 4, 5, 6$ to compute the leftmost eigenvalue of this matrix. Our initial guess space is the Krylov subspace of dimension 15 generated with the matrix $(A + 2I)^{-1}$ and initial guess vector $[1, 1, \ldots, 1]^T$. The convergence histories are shown in Figures 5-22 (a) and (b). We see that the split sizes 4, 5, and

6 show reductions in both mvs and flops against split sizes 1 and 0. This reduction is because removing the leftmost eigenvalues in this way reduces the distance between the eigenvalues of $\mathcal{Q}(A - sI)\mathcal{Q}$ and the origin. This matrix does not have outlying eigenvalues which are close together.

We remark that the number of mvs and flops plotted for ARQI (split size 0) do not match. Here the Matlab implementation of GMRES performed the maximum number of iterations but did not converge. In this case the approximate solution with smallest residual is returned and the number of iterations returned is false.

Example 5.9 The spectrum of the matrix **cavity01** is shown in Figure 5-23 (page 165). We see that the eigenvalues of the matrix are spread along the real line from 0 to 12.74.

We apply IARQI with split sizes $0, 1, 4$ to compute the rightmost eigenvalue of this matrix. Our initial guess space is the Krylov subspace of dimension 8 generated with the matrix A and initial guess vector $[1, 1, \ldots, 1]^T$. The convergence histories for $m = 0, 1, 4$ are shown in Figures 5-22 (a) and (b). We see that with split size 4 there are reductions in both mvs and flops against split sizes 1 and 0.

Example 5.10 The spectrum of the matrix **gre185** is shown in Figure 5-25 (page 166). We see that the eigenvalues of the matrix spread approximately from -0.8 to 1, the right hand section forming a large group of complex eigenvalues.

We apply IARQI with split sizes $0, 1, 4$ to compute the leftmost eigenvalue of this matrix. Our initial guess space is the Krylov subspace of dimension 10 generated with the matrix $(A+0.6I)^{-1}$ and initial guess vector $[1, 1, \ldots, 1]^T$. The convergence histories for $m = 0, 1, 4$ are shown in Figures 5-26 (a) and (b). We see that with split size 4 there is a very good reduction in mvs, and a good reduction in flops compared with split sizes 1 and 0.

5.8 Variable split sizes

We haves so far considered implementing IARQI with a fixed split size. For example, if the shift is close to two eigenvalues then it seems appropriate to use a split size $m = 2$. The following example shows that this can sometimes be counter-productive.

Example 5.11 Recall Example 5.2. We now repeat this experiment with a different initial guess space—we this time use the two dimensional Krylov subspace generated with the matrix A and the initial vector $[1, 1, \ldots, 1]^T$.

The residual norm is plotted against iterations and flops respectively in Figures 5-28 (a) and (b) (page 167) for IARQI with fixed split size $m = 2$ (labelled 2) and with a split size that was chosen based on the eigenvalue approximations available (labelled V). We see that both methods use the same number of matrix vector multiplications but that for fixed shift $m = 2$ the cost is greater.

The eigenvalue approximations available at each step of IARQI with variable split size are marked $\times$ in Figure 5-27, and the approximate eigenvalue closest to the desired eigenvalue is circled. In the first picture we see that there is only *one* approximate eigenvalue close to the desired eigenvalue. It follows that with fixed $m = 2$ we are removing an eigenvalue which is not an outlying eigenvalue—we saw in Examples 5.2, 5.3, and 5.4 that when we remove an enclosed eigenvalue we increase the cost of each mv (see 5.6) but do not reduce the number of mvs required.

Observing that there is only one approximate eigenvalue close to the desired eigenvalue we choose in the first instance to affect a split size of 1. At the next iteration there is still only one approximation to the outlying eigenvalues and we retain a split size of 1. At the next iteration we now have approximations to both outlying eigenvalues and we increase the split size to 2. In this way we minimise mvs, doing so with the minimum cost in flops.

This example shows the benefit in using a flexible split size selection strategy which takes account of the number of approximations to the outlying eigenvalues which are available. We propose the following strategy:

The (number of) eigenvalues removed at a given step of IARQI should be the (number of) approximate eigenvalues which are close to the desired eigenvalue, providing they are well separated from the rest of the spectrum.

Remark

The provision that the eigenvalues which are removed be separated from the rest of the spectrum is important. If they are not then, though there may be a reduction in mvs, the increase in the cost of an mv may cause the overall cost to increase.

In practise it is difficult to predict the overall cost—although the cost of an mv can be predicted (see Section 5.6), and although the number of iterations required by GMRES can be predicted (see Chapter 3), the cost of implementing GMRES is nonlinear and not easily evaluated.

In the following example we apply IARQI with variable split size to the matrix **fidap001**.

Example 5.12 The spectrum of the matrix **fidap001** is shown in Figure 5-29 (page 168). We see that the eigenvalues of the matrix spread along the real line approximately from -0.8 to 1.

We apply IARQI with variable split size to compute the rightmost eigenvalue of this matrix. Our initial guess space is the Krylov subspace of dimension 8 generated with the matrix A and initial guess vector $[1, 1, \ldots, 1]^T$. The convergence histories for $m = 0, 1$ and with variable split size (marked V) are shown in Figures 5-30 (a) and (b). We see that with variable split size there is a good reduction in both mvs and flops compared with split sizes 1 and 0.

5.9 Preconditioning

If the spectrum of $A - \theta I$ is, excepting a small number of eigenvalues, clustered away from the origin then GMRES within IARQI will converge in a small number of iterations. If the eigenvalues of $A - \theta I$ are not so distributed then removing eigenvalues will

not significantly increase the convergence rate of GMRES. In this situation preconditioning may be employed.

In IARQI we seek to precondition the system

$$(I - ZZ^H)(A - \theta I)(I - ZZ^H)\ \tilde{q} = -r_1.$$

This is in the same form as system (2.5) which arises in deflated Jacobi-Davidson (see Section 2.3.3). In Section 2.3.6 we discussed the method of Fokkema et al. [24] for preconditioning in deflated Jacobi-Davidson, and that method can be applied here. Briefly, If M^{-1} approximates $(A - \theta I)^{-1}$ in some way, then the application of a preconditioner of the form

$$(I - ZZ^H)M(I - ZZ^H)$$

yields the preconditioned system

$$(I - YH^{-1}Z^H)M^{-1}(A - \theta I)(I - YH^{-1}Z)\ z = (I - YH^{-1}Z)M^{-1}r,$$

where $Y = M^{-1}Z$ and $H = Z^HY$.

Example 5.13 We repeat the experiment in Example 5.12 with M arising as an incomplete LU factorisation of the shifted matrix. Figure 5-31 (page 169) shows the residual norm against the number of mvs and Figure 5-32 shows the residual norm against the number flops. We see, as expected, that IARQI with variable split size is cheaper than IARQI with split size 1. These require 58 and 65 mvs respectively. These compare with 140 and 158 mvs when preconditioning is not used.

5.10 Proof of Theorem 5.2

In this subsection we prove Theorem 5.2, which states that

> at the kth step of Algorithm 5.2 the space spanned by the columns of the matrix V_k is a Krylov subspace.

It is illuminating to first prove the following weaker Theorem.

Theorem 5.10

Let $s_1, \ldots, s_{k-1}$ be a sequence of shifts, and for given v_1 define

$$v_i = (A - s_{i-1}I)^{-1} \cdot (A - s_{i-2}I)^{-1} \cdot \cdots \cdot (A - s_1 I)^{-1} v_1, \quad i = 2, \ldots, k.$$

Then the subspace $\mathcal{K} := \langle v_1, \ldots, v_k \rangle$ is the Krylov subspace $\mathcal{K}_k(A, v_k)$.

Proof

It is easy to see that

$$v_i = (A - s_{k-1}I) \cdot (A - s_{k-2}I) \cdot \cdots \cdot (A - s_i I) v_k, \quad i = 1, \ldots, k.$$

Since Krylov subspaces are shift invariant it follows that $\mathcal{K} := \langle v_1, \ldots, v_k \rangle = \mathcal{K}_k(A, v_k)$.

□

The following example shows how the proof of Theorem 5.2 works.

Example 5.14 Let v_1 be some starting vector, and let $v_2 = (A - s_1 I)^{-1} v_1$, $v_3 = (A - s_2 I)^{-1} \left[\frac{1}{2} v_1 + \frac{1}{2} v_2\right]$.

Then $v_3 = \frac{1}{2}(A - s_2 I)^{-1} v_1 + \frac{1}{2}(A - s_2 I)^{-1}(A - s_1 I)^{-1} v_1$. Writing $\psi = (A - s_2 I)^{-1}(A - s_1 I)^{-1} v_1$ we have

$$\begin{aligned} v_1 &= (A - s_1 I)(A - s_2 I)\psi \\ v_2 &= (A - s_1 I)^{-1}(A - s_1 I)(A - s_2 I)\psi \\ &= (A - s_2 I)\psi \\ v_3 &= \frac{1}{2}(A - s_1 I)(A - s_2 I)\psi + \frac{1}{2}(A - s_2 I)\psi. \end{aligned}$$

It is clear that $\langle v_1, v_2, v_3 \rangle = \mathcal{K}_3(A, \psi)$.

Theorem 5.11

Given v_i, Let $\mathcal{K}^{(k)} = \langle v_1, \ldots, v_k \rangle$ be generated by the the recursion

For i=1, ... ,k-1

- *let* $v_{i+1} = (A - s_i I)^{-1} \left(\sum_{j=1}^{i} \alpha_{ij} v_j \right)$ *for some scalars* s_i, α_{ij}.

Then there exists $\psi \in \mathcal{K}^{(k)}$ *such that* $\mathcal{K}^{(k)}$ *is the Krylov subspace* $\mathcal{K}_k(A, \psi)$.

Proof We now prove this theorem by induction.

Suppose that $\mathcal{K}^{(l)}$ is a Krylov subspace. Then there exists $\psi \in \mathcal{K}^{(l)}$ such that $\mathcal{K}^{(l)} = \mathcal{K}_l(A, \psi)$, and we may write $v_1, \ldots, v_l$ as a linear combination of $\psi, A\psi, \ldots, A^{l-1}\psi$.

Now, for some s, $\alpha_{l1}, \ldots, \alpha_{ll}$, let

$$v_{l+1} = (A - sI)^{-1} \left(\sum_{j=1}^{l} \alpha_{lj} v_j \right).$$

Then

$$\begin{aligned} v_{l+1} &= (A - sI)^{-1} \left(\sum_{j=1}^{l} \beta_{lj} A^{j-1} \psi \right) \\ &= \sum_{j=1}^{l} \beta_{lj} A^{j-1} (A - sI)^{-1} \psi \\ &\in \mathcal{K}_{l+1}(A, \hat{\psi}) \end{aligned}$$

where $\hat{\psi} = (A - sI)^{-1}\psi$. Now note that for $j = 1, \ldots, l-1$,

$$\begin{aligned} A^j \psi &= (A - sI)(A - sI)^{-1} A^j \psi \\ &= (A - sI) A^j (A - sI)^{-1} \psi \\ &= (A - sI) A^j \hat{\psi} \\ &= A^{j+1} \hat{\psi} - s A^j \hat{\psi}. \end{aligned}$$

It follows that $A^j \psi \in \mathcal{K}_{l+1}(A, \hat{\psi})$ for $j = 1, \ldots, l-1$, and hence that $\mathcal{K}^{(l+1)} = \mathcal{K}_{l+1}(A, \hat{\psi})$, that is, that $\mathcal{K}^{(l+1)}$ is a Krylov subspace.

It is immediate that for any s and $\alpha_{11} \neq 0$ we have that $\mathcal{K}^{(1)}$ is a Krylov subspace. The result follows by mathematical induction. □

Theorem 5.2 is an immediate corollary of Theorem 5.11.

5.11 Recovering $\tilde{p}$

We now discuss how $\tilde{p}$ may be recovered from $\tilde{q}$ if necessary. This is a digression in the development of IARQI but is of interest in its own right. The following lemma shows that once the vector $\tilde{q}$ has been computed me may use $\tilde{q}$ to compute $\tilde{p}$.

Lemma 5.12

Let $\tilde{q}$ be the solution of the system (5.6). Let $\tau = [1/\xi_1, \ldots, 1/\xi_m]$ and let $\hat{X} = [\hat{x}_1, \ldots, \hat{x}_m]$. Then the system (5.7) is equivalent to

$$\mathcal{P}(A - \theta_1 I)\left[\hat{X} + \tilde{q}\tau\right]\psi = \hat{x}_1 \tag{5.12}$$

where $\psi \in \mathbb{C}^m$ satisfies $\tilde{p} = \hat{X}\psi$.

Proof

We slightly abuse our previous notation, and write $\tilde{p} = \sum_{i=1}^{m} \alpha_i \hat{x}_i$. Then writing $\hat{X} = [\hat{x}_1, \ldots, \hat{x}_m]$ and $\psi = [\alpha_1, \ldots, \alpha_m]^T$ we may rewrite this as $\tilde{p} = \hat{X}\psi$.

Recall that $\alpha = \sum_{i=1}^{m} \alpha_i/\xi_i$. Then writing $\tau = [1/\xi_1, \ldots, 1/\xi_m]$ we may rewrite this as $\alpha = \tau\psi$.

Recall also that $\tilde{q}$ satisfies

$$\mathcal{Q}(A - \theta_1 I)\mathcal{Q}\tilde{q} = -r_1$$

and that q satisfies

$$\mathcal{Q}(A - \theta_1 I)\mathcal{Q}q = -\alpha r_1.$$

Thus $q = \alpha\tilde{q}$.

Equation (5.7) may then be rewritten

$$\mathcal{P}(A-\theta_1 I)\tilde{p}+\mathcal{P}(A-\theta_1 I)\alpha\tilde{q}=\hat{x}_1.$$

Substituting $\tilde{p}=\hat{X}\psi$ and $\alpha=\tau\psi$ we have

$$\mathcal{P}(A-\theta_1 I)\hat{X}\psi+\mathcal{P}(A-\theta_1 I)\tilde{q}\tau\psi=\hat{x}_1$$

so that

$$\mathcal{P}(A-\theta_1 I)\left[\hat{X}+\tilde{q}\tau\right]\psi=\hat{x}_1.$$

□

In equation (5.12) the m-vector ψ is the only unknown, and we may solve this system for ψ. The vector $\tilde{p}$ is then easily computed since $\tilde{p}=\hat{X}\psi$.

Recall that the projection $\mathcal{P}$ is defined by $\mathcal{P}=ZZ^H$ where Z is an orthonormal matrix with the same range as $\hat{X}$. Since equation (5.12) is implicitly a system of dimension m we may replace (5.12) with the $m\times m$ system

$$\begin{aligned} Z^H(A-\theta_1 I)\left[\hat{X}+\tilde{q}\tau\right]\psi &= Z^H\hat{x}_1 \\ &= e_1. \end{aligned} \tag{5.13}$$

It is desirable to rearrange the system (5.13) to replace $\hat{X}$ with an orthonormal matrix. To do this recall that $\hat{X}=ZU$. The residuals of the columns of the matrix Z (as approximate eigenvectors of A) are the columns of the matrix $[r_1,\ldots,r_m]U$. The columns of this matrix are multiples of a single vector, and we may write $\hat{\tau}=\tau U^{-1}$. Now (5.13) becomes

$$Z^H(A-\theta_1 I)\left[Z+\tilde{q}\hat{\tau}\right]\psi=e_1.$$

This is an $m\times m$ system—the left hand side may be explicitly formed, and solved cheaply using direct methods.

The systems (5.6) and (5.13) may be solved at line (2a) of Algorithm (5.1) in place of (5.1). The approximate solution y of (5.1) is then the linear combination $y = \tilde{p} + \alpha\tilde{q}$ of the solutions to the split systems.

5.12 IARQI for large problems

Our numerical experiments illustrate the behaviour of IARQI for some small test problems. IARQI can also be applied to problems for which n is large. To simplify the discussion we assume that A is sparse with bandwidth l. We make the following observations:

(i) The principal cost in IARQI (Algorithm 5.2) is in solving

$$\mathcal{Q}(A - \theta_1 I)\mathcal{Q}y = \hat{x}_1. \tag{5.14}$$

We advocate the use of GMRES for this solve. The principal cost involved is that of repeated multiplications with $\mathcal{Q}(A - \theta_1 I)$ (see Section 5.6). Note:

- The cost of a multiplication by $(A - \theta_1 I)$ is $\mathcal{O}(2ln)$ flops.
- The cost of applying $\mathcal{Q}$ is $\mathcal{O}(4mn)$ flops (see Section 5.6).

Thus the cost of a multiplication with $\mathcal{Q}(A - \theta_1 I)$ is $\mathcal{O}((4m + 2l)n)$ flops, which is proportional to n.

The number of steps of GMRES required to solve (5.14) is independent of n (see Chapter 3), and depends upon the distribution of the eigenvalues of $\mathcal{Q}(A - \theta_1 I)\mathcal{Q}$. It follows that the cost of solving (5.14) is in proportion to n—there is no disproportional penalty incurred for large problems.

(ii) A large component of the cost of applying the Rayleigh-Ritz procedure is that of computing the eigenvalues and eigenvectors of the $k \times k$ matrix $H_k := V_k^H A V_k$. The cost of computing these eigenvalues and eigenvectors, given H_k, is approximately $\mathcal{O}(15k^3)$ (see Golub and Van Loan, page 235 [29, Sec. 7.3]), and is independent of n.

H_k may be formed in such a way that only one matrix vector multiplication with A, and only k inner products, are required. These cost $\mathcal{O}(2ln)$ flops and $\mathcal{O}(kn)$ flops respectively. These costs are in proportion to n

(iii) The costs of lines 1, 2b, and 2d in Algorithm 5.2 are also in proportion to n.

We conclude that the cost of applying IARQI is roughly proportional to the order of the matrix A, involving terms linear in n. In particular there are no disproportional costs which might make the application of IARQI to large problems impractical.

5.12.1 Restarting

The cost of the kth step of IARQI increases with k. This is because at the kth step $\tilde{q}$ must be orthonormalised against $k-1$ vectors, and the eigenvalues of a $k \times k$ matrix must be computed. At some point the cost of performing the kth step may become too high. Also, storage is required for the $n \times k$ matrix V_k. Memory limitations may force us to limit the size of k—this is particularly likely when n is large.

A common solution to these cost and memory limitations is to *restart* (see for example Saad [54]). To do this, loosely speaking, one extracts (the usually small amount of useful) information from V_k. The IARQI can then be restarted with a new initial guess space which contains this information.

The approximate eigenvectors $\hat{x}_1, \ldots, \hat{x}_m$ are used in IARQI to reduce the cost of the application of GMRES. To enable the post-restart IARQI to apply GMRES cheaply it is therefore essential that these approximate eigenvectors be made available to the restarted iteration. Thus $\langle \hat{x}_1, \ldots, \hat{x}_m \rangle$ would be an appropriate initial guess space. For such a space the initial application of the Rayleigh-Ritz procedure may be avoided since the required approximate eigenpairs usually obtained from this subspace are already known. Subspaces which contain $\langle \hat{x}_1, \ldots, \hat{x}_m \rangle$ are also suitable. This technique is called *Thick Restarting* (see Stathopoulos, Saad and Wu [70]).

Recall that when our initial guess space is generated by a shift-invert technique we can think of V_k as spanning a Krylov subspace generated by A. Consequently polynomial restarting may be applied, and in particular, implicit restarting may be

used (see Sorensen [67] and Stathopoulos et al. [70]). One must then take care that the applied restarting technique does not *remove* the vectors $\hat{x}_1, \ldots, \hat{x}_m$ which we require to be present in the new initial guess space.

5.12.2 Implementation for large problems

To implement IARQI for large problems it is necessary to use lower level languages than Matlab, for example Fortran or C. Using such languages allows computations to be performed more quickly but requires the user to implement themselves operations such as inner products and matrix vector multiplications which are implemented using single instructions in Matlab. However, to implement IARQI in Fortran or C one can take advantage of the following:

(i) Software libraries such as LAPACK (see Anderson et al. [1]) provide a number of routines which can be used, for example, to compute the eigenpairs of small matrices such as those which arise in the Rayleigh-Ritz procedure in IARQI, or to perform QR decompositions of matrices.

(ii) Routines which implement GMRES are available from Templates (see [5]). Multiplication by the iteration matrix in GMRES is performed by calling an appropriate user written subroutine. It is straightforward to incorporate within this routine the projections required by IARQI.

(iii) The only operations which involve A are matrix vector multiplications. A need not be explicitly formed and only its action on a vector is required.

5.13 Summary

We have developed a new iterative method called the Iterative Accelerated Rayleigh Quotient Iteration method (IARQI). This method generalizes the Jacobi-Davidson method but its origin is in the Accelerated Rayleigh Quotient Iteration. The convergence of IARQI is fully analysed—the method converges superlinearly, and with suitable initial guess subspaces the method is mathematically equivalent to the Ac-

celerated Rayleigh Quotient Iteration which converges quadratically (cubically if A is symmetric).

The motivation in developing IARQI was to develop a method which requires fewer steps of GMRES in each inner iteration than the Accelerated Rayleigh Quotient Iteration and the Jacobi-Davidson method. The insight given by Chapter 3 into the interaction of these methods with GMRES suggests that well chosen split sizes will produce significant reductions in the number of iterations required. We have applied the IARQI to a number of test problems and shown that this is indeed the case.

In comparison with the Jacobi-Davidson method, IARQI reduces the number of steps of GMRES required, but the cost of each of these steps is increased. Consequently comparison of the full cost of these methods is difficult. Loosely speaking, there must be a good reduction in the number of GMRES iterations for IARQI to be cheaper. Ultimately the capability of IARQI to reduce the number of GMRES iterations required depends on the distribution of the eigenvalues of A. If the convergence rate of GMRES for the shifted matrix is impaired by a small number of small eigenvalues, and would but for these converge quickly, then there is potential for great gain. For matrices which do not fit into this category the *removal* of any number of small eigenvalues will produce little gain.

We have briefly discussed adaptive techniques for selecting the split size used at each step of IARQI. With the understanding developed in Chapter 3 it is possible to develop a strategy which uses higher split sizes only when they will produce a reduction in cost. With such a strategy the Iterative ARQI will be no more expensive to implement than the Jacobi-Davidson method, and will often be cheaper.

The cost of applying IARQI can be briefly summed up as follows:

1. If the desired eigenvalue lies in a cluster of eigenvalues which is well separated from the rest of the spectrum then this eigenvalue will be computed much more cheaply using IARQI than it would using ARQI.

2. If the desired eigenvalue lies at the edge of the spectrum but is not an outlying eigenvalue then increasing the split size in IARQI will reduce the cost of computing the eigenvalue but the gains may be small.

3. If the desired eigenvalue is not separated from the remainder of the spectrum then we cannot expect IARQI to be cheaper than ARQI. In this case removing any number of eigenvalues cannot reduce the cost of applying GMRES. However, using an adaptive splitting strategy will mean that the Iterative Accelerated Rayleigh Quotient Iteration will be no more expensive than the Accelerated Rayleigh Quotient Iteration.

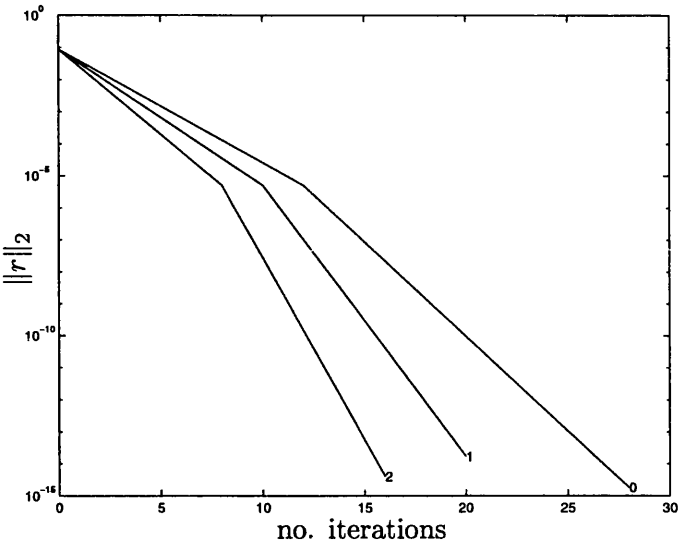


Figure 5-3: Residual norm against no. mvs for Example 5.2. Splitsizes are displayed at the end of each line.

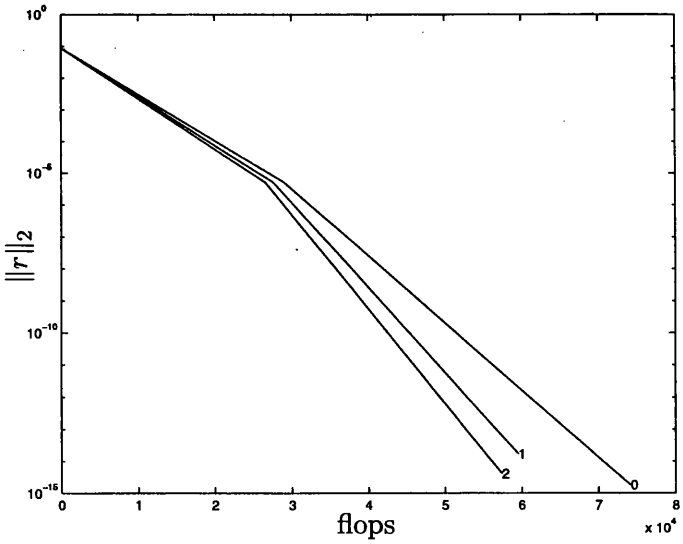


Figure 5-4: Residual norm against flops for Example 5.2. Splitsizes are displayed at the end of each line.

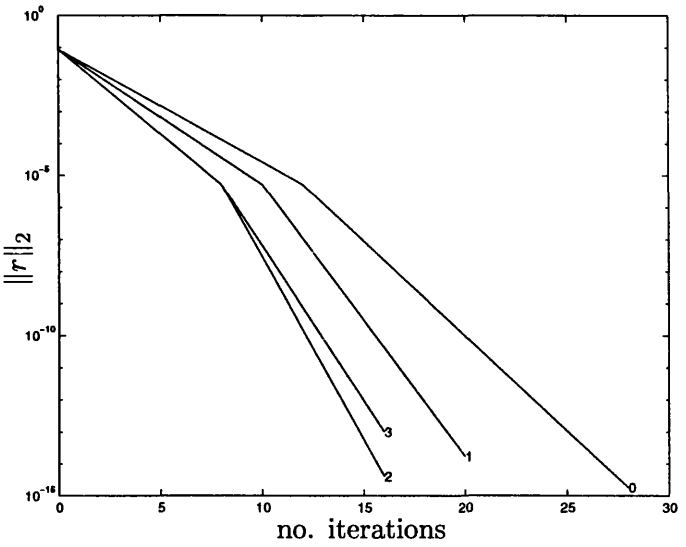


Figure 5-5: Residual norm against no. mvs for Example 5.2. Splitsizes are displayed at the end of each line.

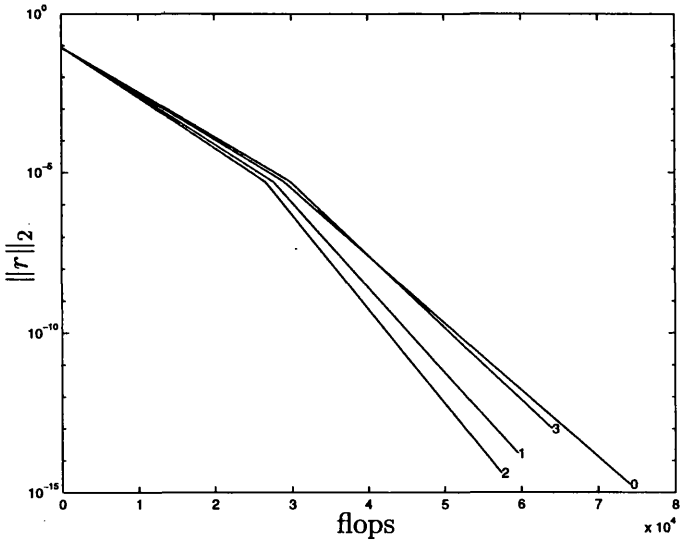


Figure 5-6: Residual norm against flops for Example 5.2. Splitsizes are displayed at the end of each line.

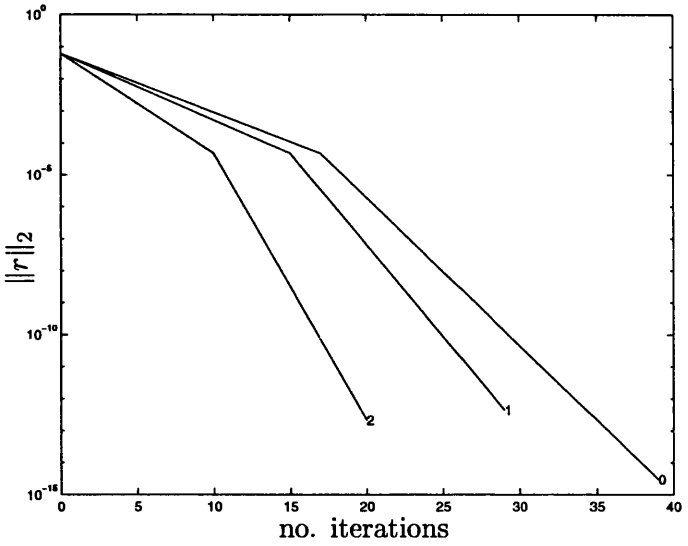


Figure 5-7: Residual norm against no. mvs for Example 5.3. Splitsizes are displayed at the end of each line.

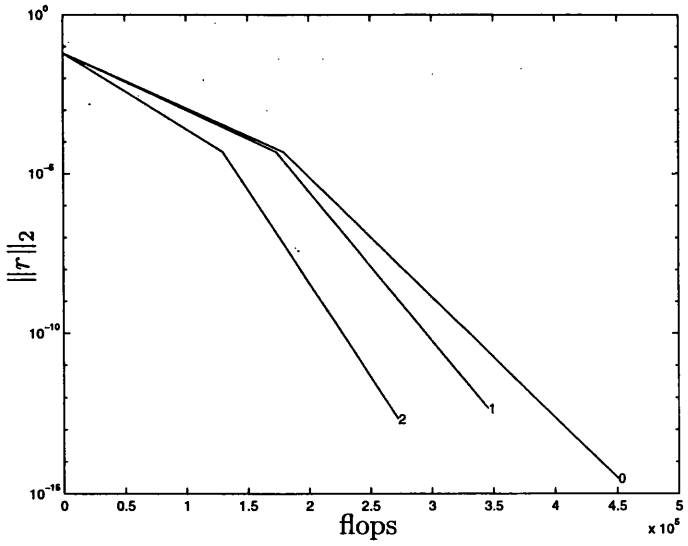


Figure 5-8: Residual norm against flops for Example 5.3. Splitsizes are displayed at the end of each line.

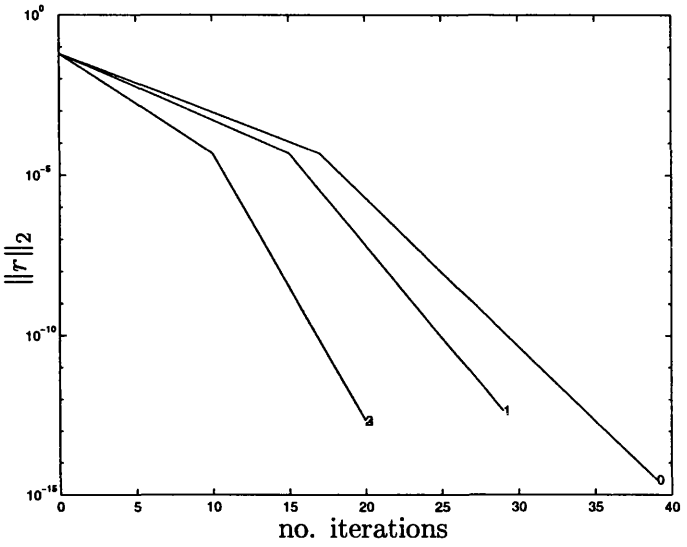


Figure 5-9: Residual norm against no. mvs for Example 5.3. Splitsizes are displayed at the end of each line.

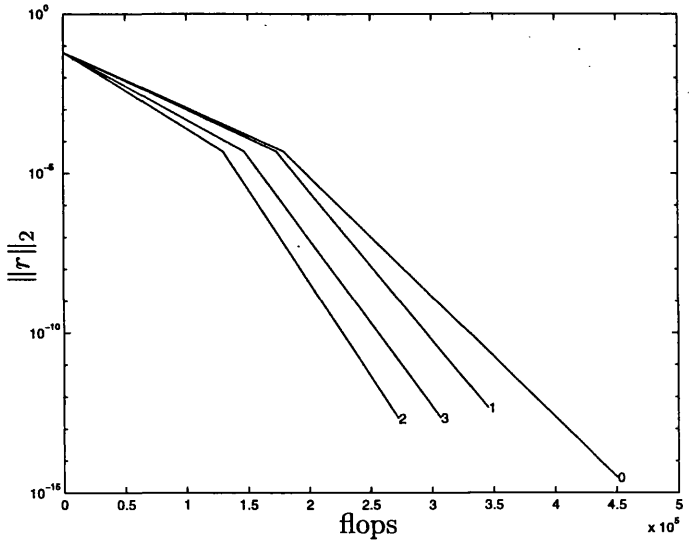


Figure 5-10: Residual norm against flops for Example 5.3. Splitsizes are displayed at the end of each line.

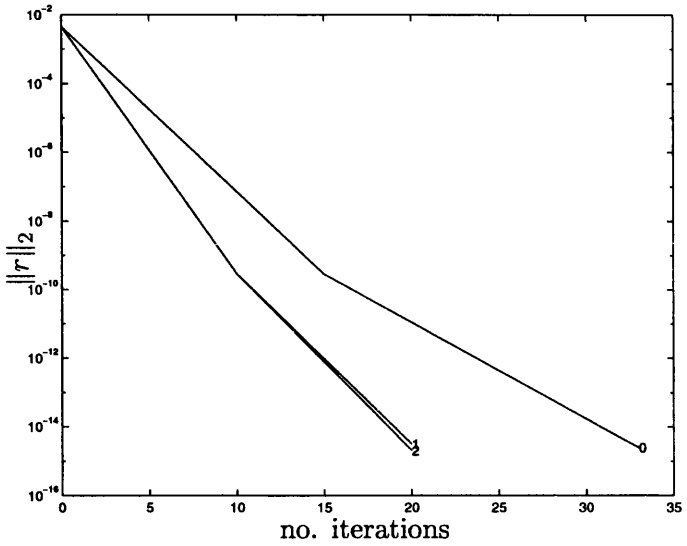


Figure 5-11: Residual norm against no. mvs for Example 5.4. Splitsizes are displayed at the end of each line.

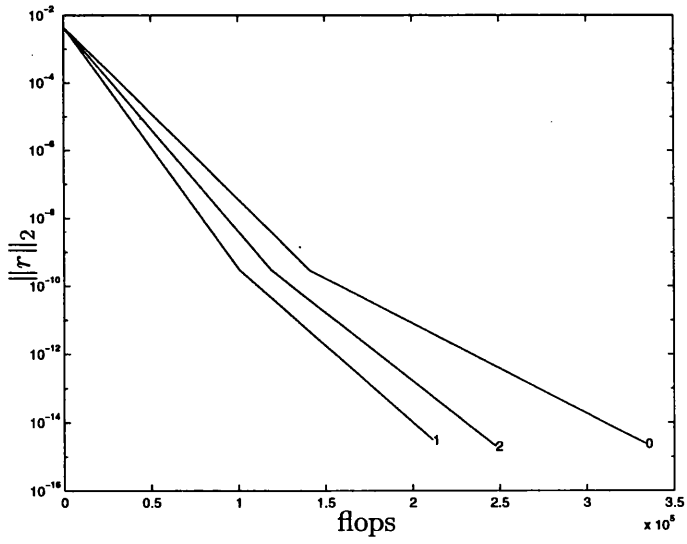


Figure 5-12: Residual norm against flops for Example 5.4. Splitsizes are displayed at the end of each line.

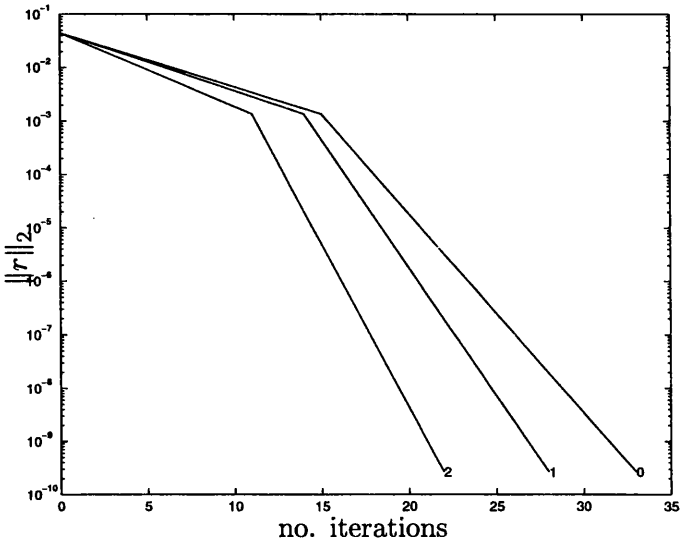


Figure 5-13: Residual norm against no. mvs for Example 5.5. Splitsizes are displayed at the end of each line.

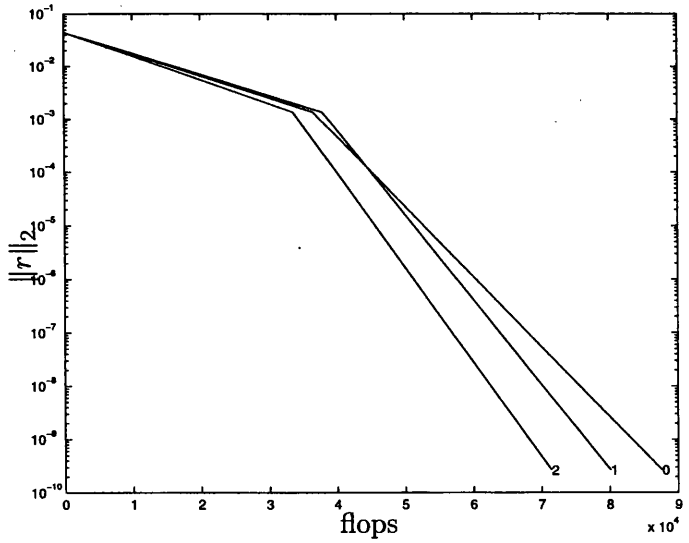


Figure 5-14: Residual norm against flops for Example 5.5. Splitsizes are displayed at the end of each line.

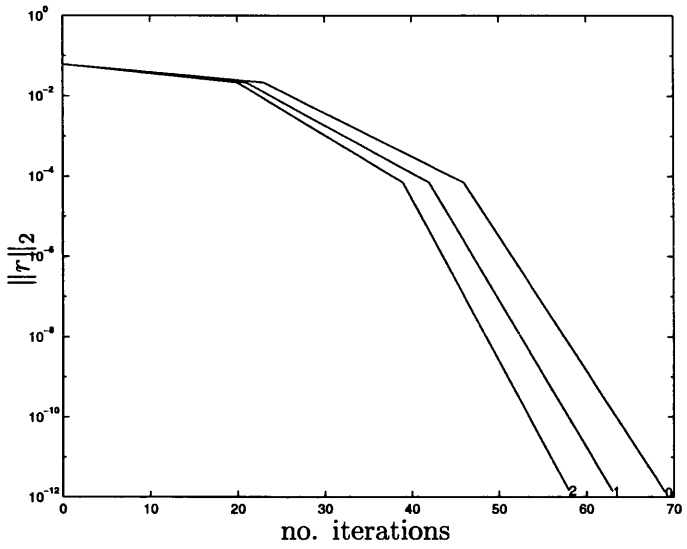


Figure 5-15: Residual norm against no. mvs for Example 5.5. Splitsizes are displayed at the end of each line.

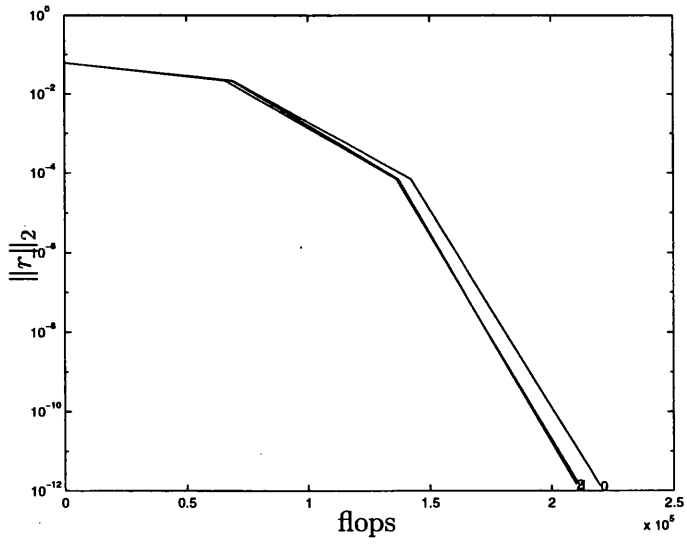


Figure 5-16: Residual norm against flops for Example 5.5. Splitsizes are displayed at the end of each line.

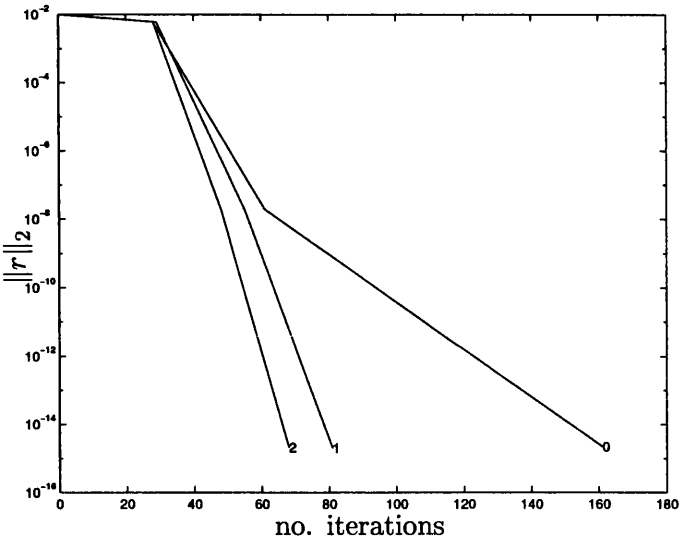


Figure 5-17: Residual norm against no. mvs for Example 5.6. Splitsizes are displayed at the end of each line.

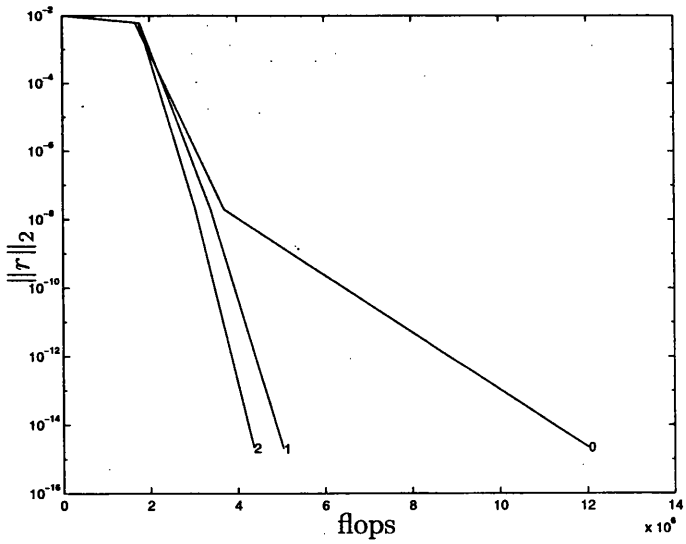


Figure 5-18: Residual norm against flops for Example 5.6. Splitsizes are displayed at the end of each line.

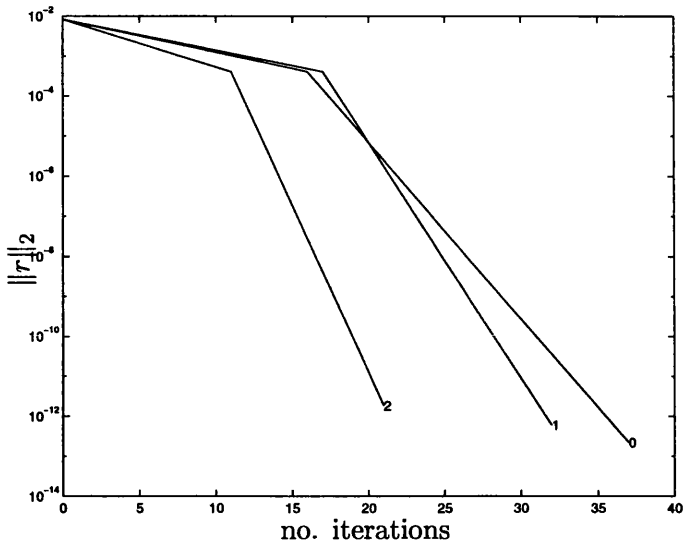


Figure 5-19: Residual norm against no. mvs for Example 5.7. Splitsizes are displayed at the end of each line.

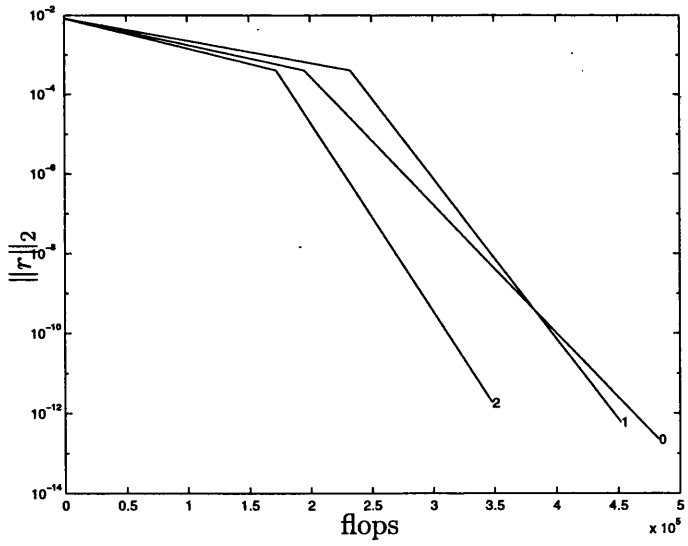


Figure 5-20: Residual norm against flops for Example 5.7. Splitsizes are displayed at the end of each line.

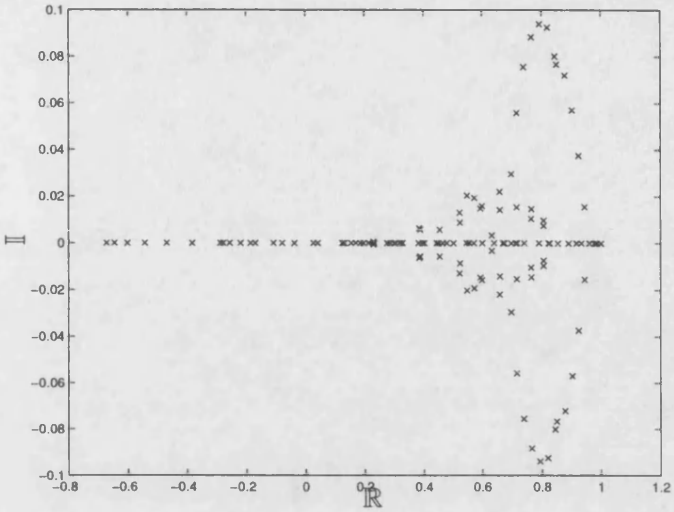


Figure 5-21: Spectrum of the matrix **lop163**.

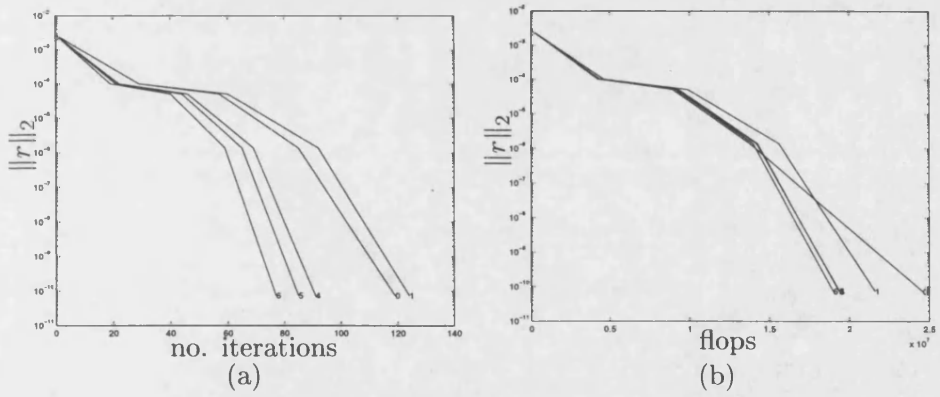


Figure 5-22: Residual norm against (a) mvs and (b) flops for Example 5.8. Splitsizes are displayed at the end of each line.

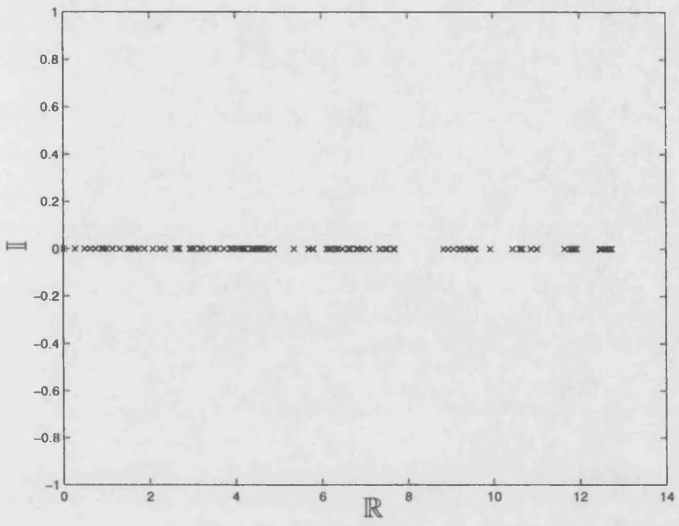


Figure 5-23: Spectrum of the matrix **cavity01**.

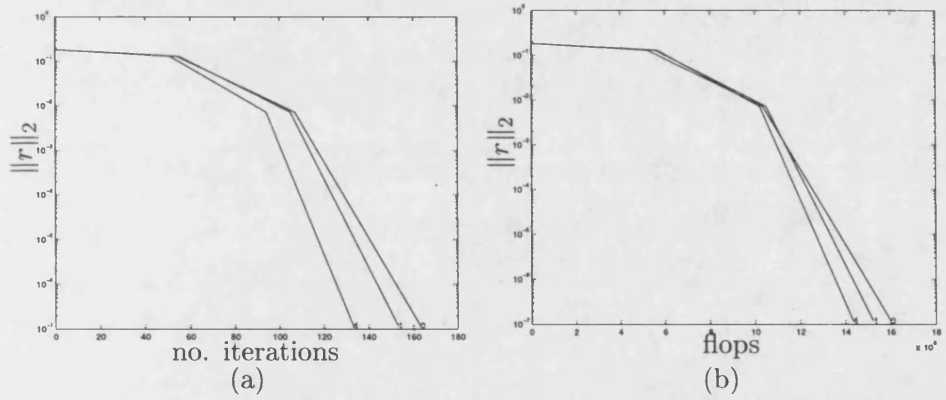


Figure 5-24: Residual norm against (a) mvs and (b) flops for Example 5.9. Splitsizes are displayed at the end of each line.

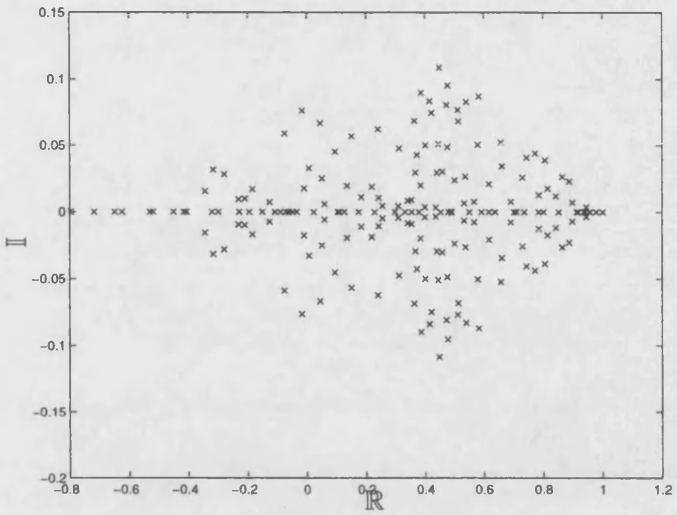


Figure 5-25: Spectrum of the matrix **gre185**.

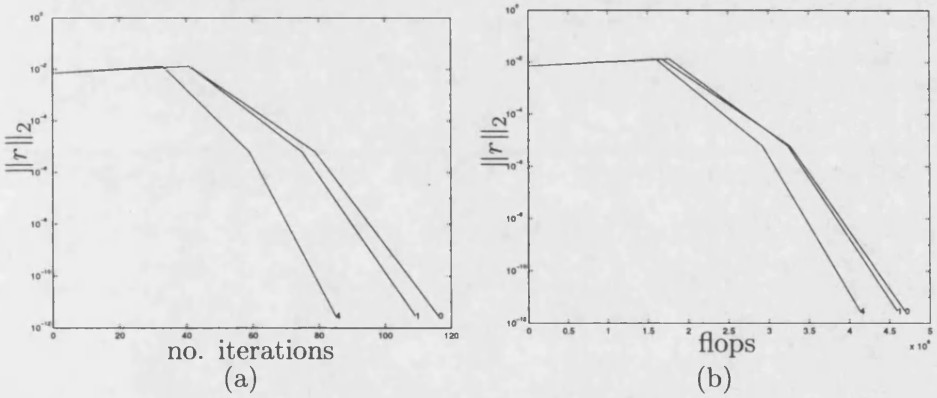


Figure 5-26: Residual norm against (a) mvs and (b) flops for Example 5.10. Splitsizes are displayed at the end of each line.

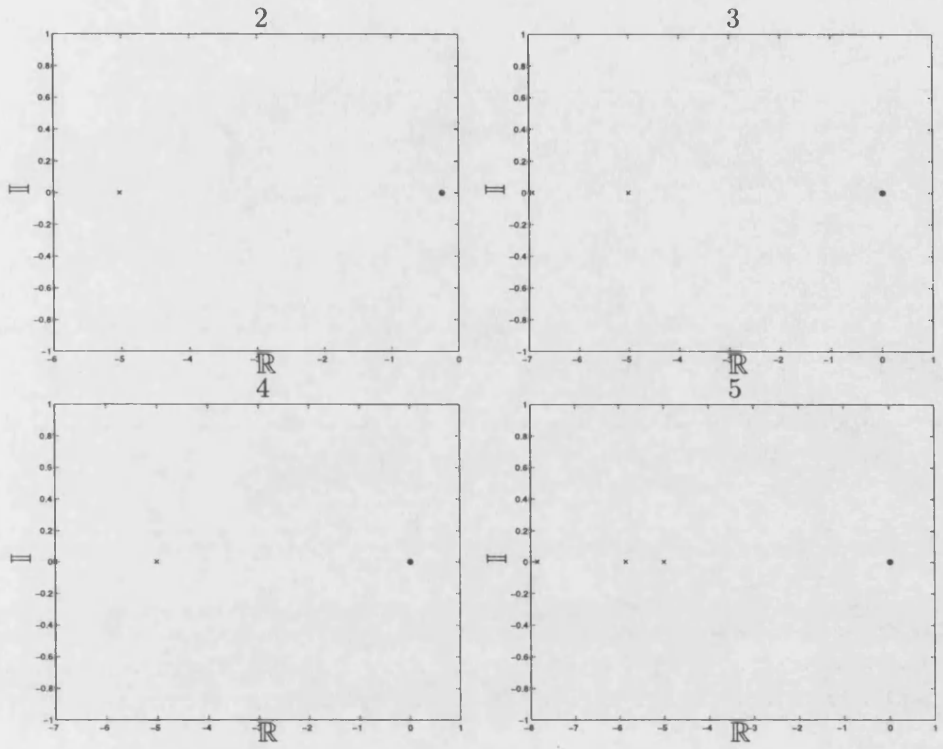


Figure 5-27: Eigenvalue approximations from subspaces of size 2, 3, 4, 5 in Example 5.11

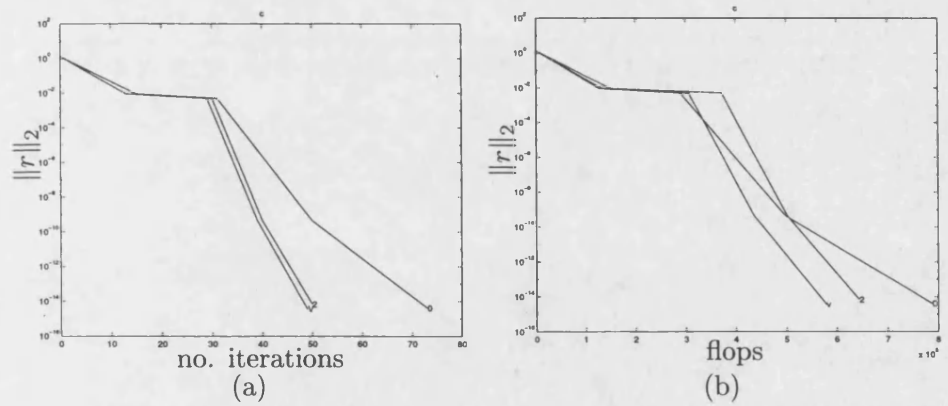


Figure 5-28: Residual norm against (a) mvs and (b) flops for Example 5.11. Splitsizes are displayed at the end of each line.

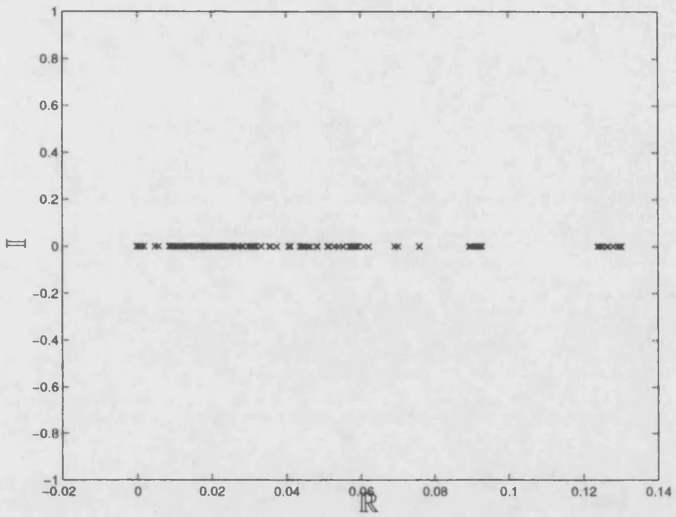


Figure 5-29: Spectrum of the matrix **fidap001**.

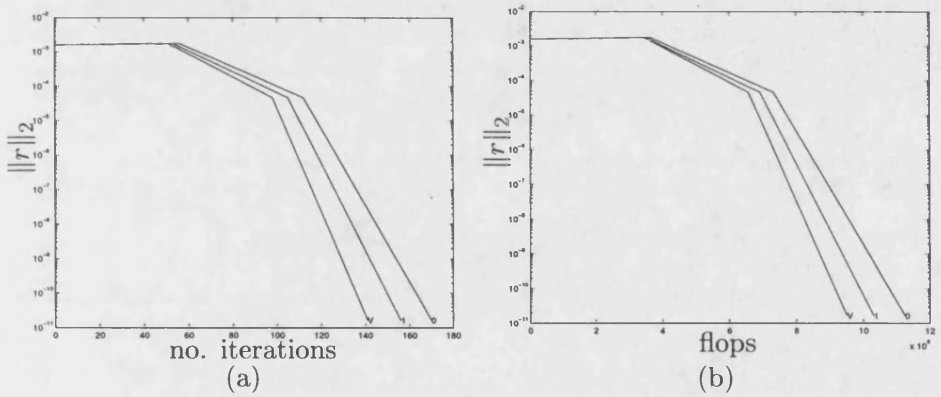


Figure 5-30: Residual norm against (a) mvs and (b) flops for Example 5.12. Splitsizes are displayed at the end of each line.

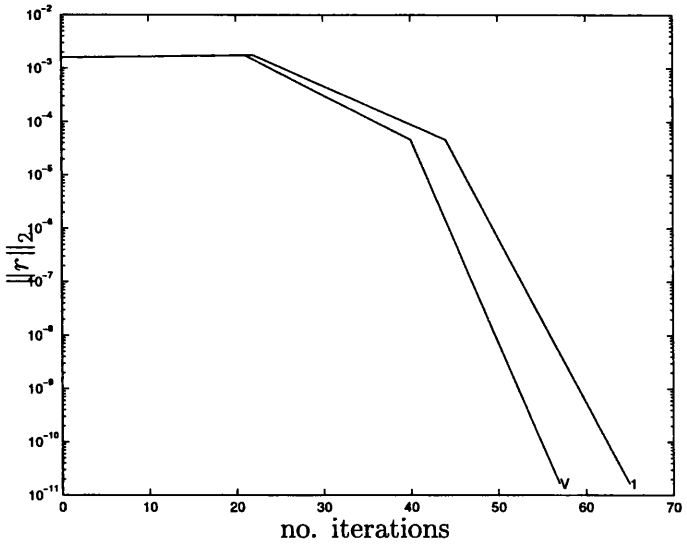


Figure 5-31: Residual norm against no. mvs for Example 5.13. Splitsizes are displayed at the end of each line.

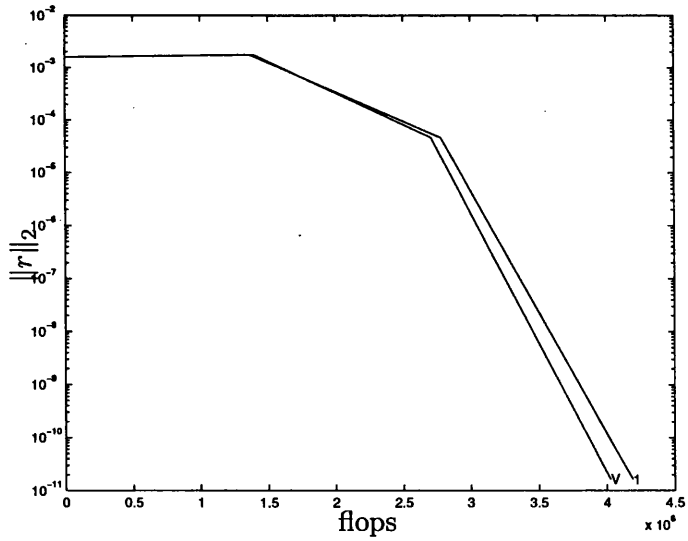


Figure 5-32: Residual norm against flops for Example 5.13. Splitsizes are displayed at the end of each line.

Chapter 6

Detecting Hopf Bifurcations

6.1 Introduction

For an ordinary differential equation (ODE) our interest falls into three general categories: the behaviour of the solution over time, the steady states (or general limit sets) of the ODE, and the stability of any steady states of the ODE. The tools used for studying the behaviour of an ODE in each of these categories are often impaired when the ODE is stiff. To compute steady states Mamun and Tuckerman [37] and Davidson [15] employ preconditioning in Newton's method and the Recursive Projection Method ([61]) respectively to combat stiffness and reduce cost. Often the same tools used for time integration and steady state solving can be used to give information about the spectrum of the preconditioned Jacobian, but in general the preconditioner transforms the spectrum of the Jacobian in a complicated way which we cannot analytically invert.

For many ODEs that arise as spatial discretizations of semilinear PDEs the *Stokes preconditioner* [4] is the natural choice of preconditioner. In this chapter we present two techniques which use the spectrum of the Stokes preconditioned Jacobian to approximate the spectrum of the Jacobian. The spectrum of the Jacobian can then be used to detect bifurcation. In particular we are interested in detecting Hopf bifurcation—we will see in Section 6.2 that this is a more challenging task than detecting steady bifurcations.

The outline of this chapter is as follows. We begin Section 6.2 with a description

of our general parameter dependent semilinear partial differential equation. In Section 6.3 we show that the eigenvalues of the Stokes preconditioned Jacobian are continuous functions of the parameter ω in the Stokes preconditioner. We then show that given eigeninformation for the preconditioned Jacobian it is possible to correct its eigenvalues to obtain second order approximations to eigenvalues of the Jacobian itself. In Section 6.3.1 we apply these results to the Tubular Reactor problem. In Section 6.4 we extend the approach of Section 6.3 to show that the critical point at which the Stokes preconditioned Jacobian has pure imaginary rightmost eigenvalues is itself a function of ω. From this we develop a method for approximating a Hopf bifurcation point by correcting from the critical point for the Stokes preconditioned Jacobian. In Section 6.4.1 we illustrate these results for the Tubular Reactor problem.

6.2 Linear stability analysis

Consider the general semilinear partial differential equation

$$u_t = \mathcal{L}u + \mathcal{N}(u, \lambda) \tag{6.1}$$

where $\mathcal{L}$ is a linear operator and $\mathcal{N}$ is a nonlinear mapping which is dependent upon the parameter $\lambda \in \mathbb{R}$. We assume that we may spatially discretize (6.1) to obtain the n-dimensional system of ordinary differential equations

$$U_t = LU + N(U, \lambda), \tag{6.2}$$

where L is an $n \times n$ real matrix and $N : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is a smooth function of U and λ. Typically $\mathcal{L}$ is an operator such as the Laplacian.

To determine the linear stability of a steady state U of (6.2) we must compute the (rightmost) eigenvalues of the Jacobian matrix

$$A := L + N_U(U, \lambda).$$

We would prefer to determine linear stability from the eigenvalues of a preconditioned

Jacobian instead of eigenvalues of A. This is because

- Typically the spectrum of the matrix A is dominated by the spectrum of L, and so A is badly conditioned. Preconditioning can produce a well conditioned matrix whose eigenvalues are easier to compute, for example using Inverse Iteration (see Section 1.3.2).
- Often the eigenvalues of a preconditioned Jacobian are obtained for free when steady state solving, for example in the Preconditioned Recursive Projection Method (Davidson [15]).

In Barkley and Tuckerman [4], Davidson [15], and Mamun and Tuckerman [37], the matrix $(I - \Delta t L)$ is used to precondition A. The preconditioned matrix

$$(I - \Delta t L)^{-1}(L + N_U(U, \lambda))$$

is typically better conditioned than A. For example, Figure 6-1 shows the spectra of A and of $(I - \Delta t L)^{-1}(L + N_U(U, \lambda))$ for the Tubular Reactor problem with $\Delta t = 0.1$ (see Section 6.3.1). The rightmost eigenvalues of the preconditioned Jacobian are much easier to compute than those of the Jacobian itself—Figure 6-2 shows the convergence history for Arnoldi's method applied to these matrices. Arnoldi's method computes the rightmost eigenvalues of $(I - 10^{-1}L)^{-1}(L + N_U(U, \lambda))$ in less than 20 steps, and only computes the eigenvalues of $L + N_U(U, \lambda)$ at the 200th step, that is, when they are computed directly using the QR method. The cost of multiplying a vector by $(I - \Delta t L)^{-1}(L + N_U(U, \lambda))$ is greater than the cost of multiplying by $L + N_U(U, \lambda)$ but considering residual norm against cpu time we still see a substantial reduction in cost (see Figure 6-3). Furthermore, it is shown by Mamun and Tuckerman [37] that the action of $(I - \Delta t L)^{-1}(L + N_U(U, \lambda))$ can be easily obtained by using time stepping codes which implement a Forward Euler/Backward Euler (FEBE) scheme for (6.2).

Davidson [15] gives the following result.

Proposition 6.1 (Davidson [15, Proposition 2])

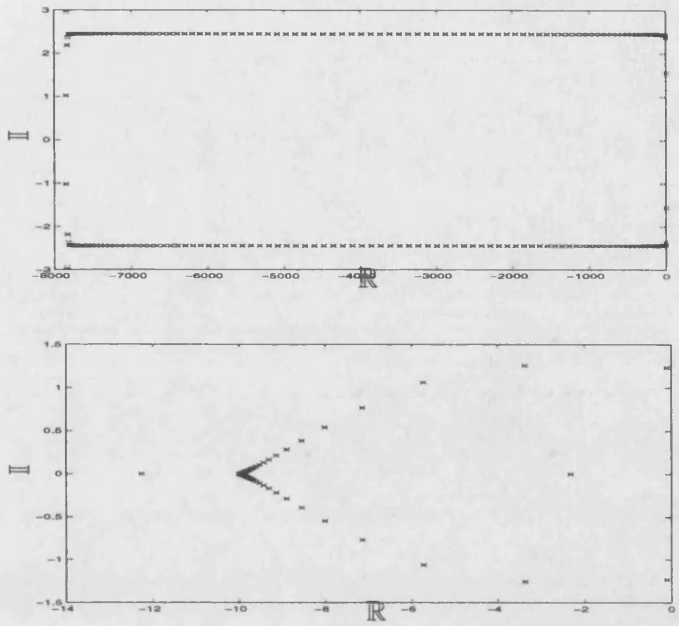


Figure 6-1: Spectrum of $L+N_U(U,\lambda)$ (top) and $(I-10^{-1}L)^{-1}(L+N_U(U,\lambda))$ (bottom).

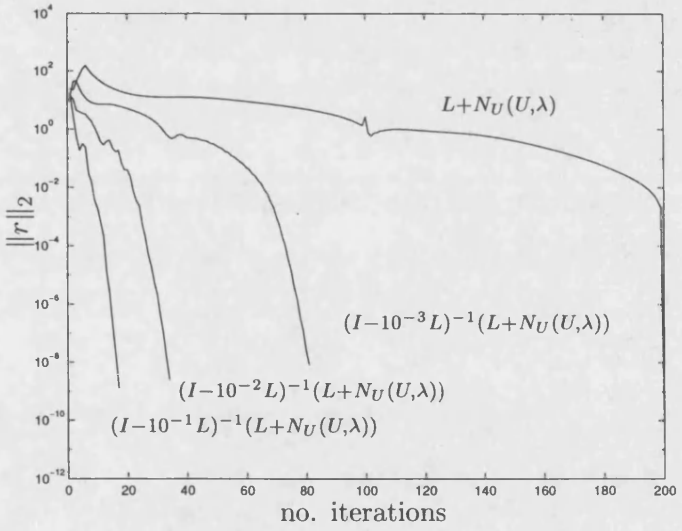


Figure 6-2: Convergence history for Arnoldi's method applied to the Jacobian and to some preconditioned Jacobians for the Tubular Reactor problem.

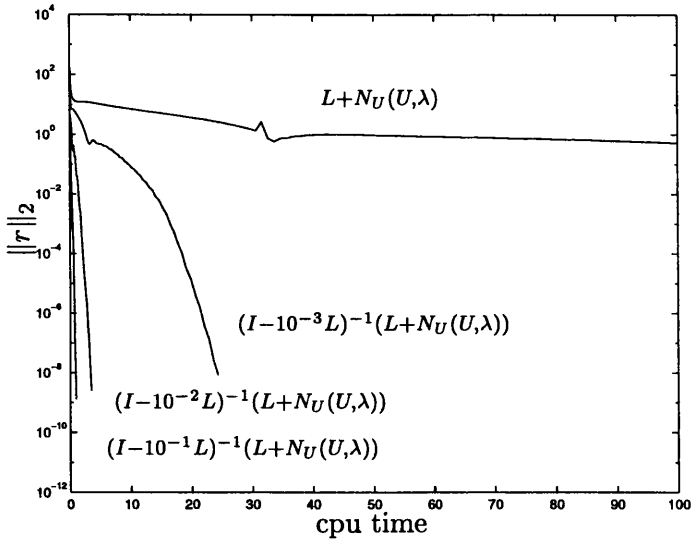


Figure 6-3: Convergence history for Arnoldi's method applied to the Jacobian and to some preconditioned Jacobians for the Tubular Reactor problem.

Consider the two eigenvalue problems

$$\begin{aligned}(L + N_U(U,\lambda))x_i &= \nu_i x_i, \quad i = 1,\ldots,n,\\ (I - \Delta t L)^{-1}(L + N_U(U,\lambda))y_j &= \theta_j y_j, \quad j = 1,\ldots,n.\end{aligned}$$

Then $\nu_i = 0$ implies that there exists an index j for which $\theta_j = 0$, and $\theta_j = 0$ implies that there exists an index i for which $\nu_i = 0$.

Consequently, if there is a change in stability where A has a zero eigenvalue, then this change in stability will be indicated by a zero eigenvalue of $(I - \Delta t L)^{-1}(L + N_U(U,\lambda))$. When the stability changes at a Hopf bifurcation point A has a pair of pure imaginary eigenvalues, but the preconditioned matrix may not have pure imaginary eigenvalues. How can we detect a Hopf bifurcation using the eigenvalues of the preconditioned matrix?

We begin by noting that the problem of computing eigenvalues of the preconditioned problem resolves to solving the n dimensional generalised eigenvalue problem

$$A(\lambda)x = \theta B(\omega)x, \quad x \in \mathbb{C}^n,\ \theta \in \mathbb{C}$$

where A is an $n \times n$ real matrix dependent on the real parameter λ, B is an $n \times n$ real matrix depending on a second real parameter ω. We shall assume that the complex eigenvector x is normalised by $c^H x = 1$ for a suitably chosen constant complex vector c.

Although we are interested in computing the eigenvalues of the Stokes preconditioned Jacobian we present the analysis in this chapter for the general case, and we assume

(A1) $A(\lambda)$ and $B(\omega)$ are C^k, $k \geq 2$, functions of λ and ω respectively.

(A2) $y_0 := (x_0, \theta_0) \in \mathbb{C}^{n+1}$ is an algebraically simple eigenpair of $A(\lambda_0)x_0 = \theta_0 B(\omega_0)x_0$.

(A3) $c^H x_0 = 1$.

Let v_0 be the left null vector of $A(\lambda_0) - \theta_0 B(\omega_0)$, so that

$$v_0^T (A(\lambda_0) - \theta_0 B(\omega_0)) = 0. \tag{6.3}$$

Now (A2) implies that $B(\omega_0)x_0 \notin \text{range}(A(\lambda_0) - \theta_0 B(\omega_0))$, and hence

$$v_0^T B(\omega_0) x_0 \neq 0. \tag{6.4}$$

We shall use the notation $\theta_\omega^0 = \frac{d}{d\omega}(\theta(\omega_0, \lambda_0))$ etc.

We now consider two approaches for detecting a Hopf bifurcation point using eigenvalues of the preconditioned matrix:

Approach 1 We approximate the rightmost eigenvalues of A by correcting from the rightmost eigenvalues of the preconditioned matrix.

Approach 2 We approximate the Hopf bifurcation point where A has pure imaginary rightmost eigenvalues by correcting from the point where the preconditioned matrix has pure imaginary rightmost eigenvalues.

6.3 Approach 1: Eigenvalue correction

Consider the problem of determining stability of a fixed point U for fixed parameter λ. Stability may be determined by computing the (rightmost) eigenvalues of the Jacobian $A(\lambda) = L + N_U(U, \lambda)$. We begin by comparing the eigenvalues of $A(\lambda)$ and the preconditioned matrix $B(\omega)^{-1}A(\lambda)$ as ω varies.

Theorem 6.2 *Assume (A1), (A2) and (A3). Then*

(i) there exists a neighbourhood $V \subseteq \mathbb{R}^2$ of (ω_0, λ_0) such that for $(\omega, \lambda) \in V$, $y := (x, \theta)$ is an algebraically simple eigenpair of $A(\lambda)x = \theta B(\omega)x$. Furthermore, $y(\omega_0, \lambda_0) = y_0$, and $y = y(\omega, \lambda)$ is a C^k, $k \geq 2$, function.

(ii)

$$\theta_\omega(\omega_0, \lambda_0) = -\theta_0 \frac{v_0^T B_\omega(\omega_0) x_0}{v_0^T B(\omega_0) x_0},$$

$$\theta_\lambda(\omega_0, \lambda_0) = \frac{v_0^T A_\lambda(\lambda_0) x_0}{v_0^T B(\omega_0) x_0},$$

where the denominator is nonzero by (6.4).

(iii) the eigenvalue $\theta(\omega, \lambda)$ satisfies the Taylor expansion

$$\theta(\omega, \lambda) - \theta(\omega_0, \lambda_0) = (\omega - \omega_0)\theta_\omega^0 + (\lambda - \lambda_0)\theta_\lambda^0 + h.o.t. \tag{6.5}$$

Proof

(i) Define $F : \mathbb{C}^{n+1} \times \mathbb{R}^2 \to \mathbb{C}^{n+1}$ by

$$F(y, \omega, \lambda) = \begin{bmatrix} A(\lambda)x - \theta B(\omega)x \\ c^H x - 1 \end{bmatrix}, \quad y = \begin{bmatrix} x \\ \theta \end{bmatrix}.$$

Clearly y is a solution of $A(\lambda)x = \theta B(\omega)x$ with $c^H x = 1$ if and only if $F(y, \omega, \lambda) =$

0. Then $F(y_0, \omega_0, \lambda_0) = 0$ and

$$F_y(y_0, \omega_0, \lambda_0) = \begin{bmatrix} A(\lambda_0) - \theta_0 B(\omega_0) & -B(\omega_0)x_0 \\ c^H & 0 \end{bmatrix}$$

with $\text{rank}(A(\lambda_0) - \theta_0 B(\omega_0)) = n - 1$.

The ABCD Lemma (Lemma A.2) is now used to prove that this matrix is nonsingular. Since $\ker(A(\lambda_0) - \theta_0 B(\omega_0)) = \text{span}\{x_0\}$, we have $c^H \phi \neq 0$ for all $\phi \in \ker(A(\lambda_0) - \theta_0 B(\omega_0)) \setminus \{0\}$. Also, using (6.4), $\psi^T(-B(\omega_0)x_0) \neq 0$ for all $\psi \in \ker(A(\lambda_0) - \theta_0 B(\omega_0))^T \setminus \{0\}$. Thus $F_y(y_0, \omega_0, \lambda_0)$ is nonsingular by Lemma A.2, part (*ii*).

Now one can apply the Implicit Function Theorem (Theorem A.1) to show the existence of $y = y(\omega, \lambda)$, a unique solution of $A(\lambda)x = \theta B(\omega)x$ for $(\omega, \lambda) \in V$. This shows that $x = x(\omega, \lambda)$, and a similar argument shows the existence of a smooth left eigenvector $v^T(\omega, \lambda)$ for $(\omega, \lambda) \in V$.

Another consequence of Theorem A.1 is that $F_y(y(\omega, \lambda), \omega, \lambda)$ is nonsingular for $(\omega, \lambda) \in V$ and hence $\text{rank}(A(\lambda) - \theta(\omega, \lambda)B(\omega)) = n - 1$, using Lemma A.2, part (*iii*). Thus $v^T(\omega, \lambda)B(\omega)x(\omega, \lambda) \neq 0$ for $(\omega, \lambda) \in V$, and so (x, θ) is an algebraically simple eigenpair of $A(\lambda) - \theta B(\omega)$ for $(\omega, \lambda) \in V$. Finally $y \in C^k, k \geq 2$.

(ii) Along a path of solutions of $F(y, \omega, \lambda) = 0$ we have

$$0 = \frac{dF}{d\omega} = F_y \, y_\omega + F_\omega,$$

that is,

$$\begin{bmatrix} A(\lambda) - \theta B(\omega) & -B(\omega)x \\ c^H & 0 \end{bmatrix} \begin{bmatrix} x_\omega \\ \theta_\omega \end{bmatrix} = - \begin{bmatrix} -\theta B_\omega(\omega)x \\ 0 \end{bmatrix}.$$

Evaluating at $(\omega, \lambda) = (\omega_0, \lambda_0)$, and multiplying out gives

$$(A(\lambda_0) - \theta_0 B(\omega_0))x_\omega^0 - \theta_\omega^0 B(\omega_0)x_0 = \theta_0 B_\omega(\omega_0)x_0.$$

The expression for θ_ω^0 follows on left multiplication by v_0^T. The proof for θ_λ^0 is similar.

(iii) This is the Taylor series under (A.1) for $\theta(\omega, \lambda)$ expanded about (ω_0, λ_0).

□

We now apply Theorem 6.2 to our Stokes preconditioned eigenvalue problem. We assume that $\lambda = \lambda_0$ is fixed and for simplicity write $\theta = \theta(\omega)$ in place of $\theta = \theta(\omega, \lambda_0)$. In addition, we consider the eigenvalue problem

$$[L + N(\lambda)]x(\Delta t) = \mu(\Delta t)[I - \Delta t L]x(\Delta t)$$

at $\omega = \Delta t$.

Corollary 6.3 *Assume that $\lambda = \lambda_0$ is fixed. Let $(x(\Delta t), \theta(\Delta t))$ be an algebraically simple eigenpair of $(I - \Delta t L)^{-1}(L + N(\lambda_0))$. Then $L + N(\lambda_0)$ has an algebraically simple eigenpair $(x(0), \theta(0))$ with $\theta(0) = \hat{\theta}(0) + \mathcal{O}(\Delta t^2)$ where*

$$\hat{\theta}(0) := \theta(0) + \Delta t\, \theta(\Delta t)\, \frac{v(\Delta t)^T L x(\Delta t)}{v(\Delta t)^T (I - \Delta t L) x(\Delta t)}. \tag{6.6}$$

Here $v(\Delta t)$ is the left null vector of $[L + N(\lambda_0) - \theta(\Delta t)(I - \Delta t L)]$.

The ability to approximate the eigenvalues of A given the eigenvalues of $(I - \Delta t L)^{-1}(L + N_U(U, \lambda))$ allows us to detect changes in stability of the steady state U. We propose the following approach:

Continue along a path of steady states (using, for example, some predictor-corrector scheme). At each step

(i) compute one of the rightmost eigenvalues $\theta(\Delta t)$ of

$$(I - \Delta t L)^{-1}(L + N(U, \lambda))$$

and the vectors $v(\Delta t)$, $x_1(\Delta t)$ described above.

(ii) compute the approximation

$$\widehat{\theta}(0) = \theta(\Delta t) - \Delta t\, \theta(\Delta t)\, \frac{v(\Delta t)^T L x(\Delta t)}{v(\Delta t)^T [(I - \Delta t L)] x(\Delta t)}.$$

(iii) look for $\widehat{\theta}(0)$ crossing the imaginary axis.

6.3.1 Numerical results for Approach 1

We now give results which demonstrate the effectiveness of the correction technique of Approach 1.

The Tubular Reactor problem We present results obtained for the Tubular Reactor problem, which is studied in detail in Heinemann and Poore [31]. The coupled equations modelling the temperature ϑ and the concentration y of reactant in a tubular reactor are

$$\begin{aligned} y_t &= \frac{1}{P_{e_m}} y_{xx} - y_x - D_a y e^{\gamma(1-\frac{1}{\vartheta})}, \\ \vartheta_t &= \frac{1}{P_{e_h}} \vartheta_{xx} - \vartheta_x - \beta(\vartheta - \vartheta_c) + B D_a y e^{\gamma(1-\frac{1}{\vartheta})}. \end{aligned}$$

These equations have linear part $(1/P_{e_m})y_{xx}$ and $(1/P_{e_h})\vartheta_{xx}$ and are of the form (6.1). We also apply boundary conditions

$$y_x(t,0) = P_{e_m}(y(t,0) - 1), \qquad \vartheta_x(t,0) = P_{e_h}(\vartheta(t,0) - 1),$$

$$y_x(t,1) = \vartheta_x(t,1) = 0, \qquad \vartheta(0,x) = \vartheta_0, \qquad y(0,x) = y_0 \qquad x \in [0,1].$$

Δt	$\lvert\widehat{\theta}(0) - \theta(0)\rvert$	$\lvert\mathrm{Re}(\widehat{\theta}(0) - \theta(0))\rvert$
10^{-1}	0.05117209364086	0.00750699571920
10^{-2}	0.00071366699203	0.00013596478198
10^{-3}	0.00000740094882	0.00000144593687
10^{-4}	0.00000007428074	0.00000001454876

Table 6.1: Error in the real part of $\widehat{\theta}(0)$ computed by correcting one of the rightmost eigenvalues $\theta(\Delta t)$ of $(I - \Delta t L)^{-1}(L + N_U(U, \lambda))$.

We consider this problem with the constants $P_{e_m} = P_{e_h} = 5$, $B = 0.5$, $\gamma = 25$ and $\beta = 3.5$. The parameter $D_a =: \lambda$ is the Damköhler number. In the tubular reactor problem we are interested in the dynamics of the system as λ changes.

A bifurcation diagram for this problem is given in Heinemann and Poore [31, Fig.1]. There is a Hopf bifurcation point at $\lambda = 0.2612274$. Table 6.1 shows the error in the real part of the approximation $\widehat{\theta}(0)$ to $\theta(0)$ at $\lambda = 0.2612274$. It is clear that this error is of order Δt^2 and that the results agree with the theory.

6.4 Approach 2: Bifurcation point correction

We now move on to the problem of determining the critical value $\lambda = \lambda_c$ at which the (rightmost) eigenvalues of $A(\lambda)$ are pure imaginary, that is, the Hopf bifurcation point.

In fact, there exists a path of pure imaginary eigenvalues of the generalised eigenvalue problem $A(\lambda)x = \theta B(\omega)x$ for ω in some neighbourhood of $\omega = 0$, and the critical value of λ at which these occur is a smooth function of ω. This is proved in the following theorem.

Theorem 6.4 *Assume (A1), (A2), (A3) and*

$$\begin{aligned} \mathrm{Re}\,(\theta(\omega_0, \lambda_0)) &= 0 \\ \mathrm{Re}\,(\theta_\lambda(\omega_0, \lambda_0)) &\neq 0. \end{aligned}$$

Then

(i) there exists a neighbourhood $V \subseteq \mathbb{R}$ *of* ω_0 *and a unique* $\lambda_c = \lambda_c(\omega)$ *such that*

$$\mathrm{Re}\,(\theta(\omega, \lambda_c(\omega))) = 0, \quad \omega \in V.$$

Furthermore $\lambda_c \in C^k$, *and* $\lambda_c(0) = \lambda_0$.

(ii)

$$\lambda_c(\omega) - \lambda_0 = -(\omega - \omega_0)\frac{\mathrm{Re}\,(\theta_\omega^0)}{\mathrm{Re}\,(\theta_\lambda^0)} + h.o.t.$$

Proof

(i) Define $G : \mathbb{C}^{n+1} \times \mathbb{R} \times \mathbb{R} \to \mathbb{C}^{n+1} \times \mathbb{R}$ by

$$G(y, \omega, \lambda) = \begin{bmatrix} A(\lambda)x - \theta B(\omega)x \\ c^H x - 1 \\ \mathrm{Re}\,(\theta) \end{bmatrix}.$$

Then

$$G_{y,\lambda}(y_0, \omega_0, \lambda_0) = \left[\begin{array}{cc|c} & & A_\lambda(\lambda_0)x_0 \\ & F_y(y_0, \omega_0, \lambda_0) & 0 \\ \hline 0^T & \mathrm{Re}\,(\cdot) & 0 \end{array}\right].$$

To prove that $G_{y,\lambda}(y_0, \omega_0, \lambda_0)$ is nonsingular we use part (i) of Lemma A.2. We showed in Theorem 6.2 that $F_y(y_0, \omega_0, \lambda_0)$ is nonsingular.

It remains to show that

$$\begin{bmatrix} 0^T & \mathrm{Re}\,(\cdot) \end{bmatrix}^H F_y^{-1}(y_0, \omega_0, \lambda_0) \begin{bmatrix} A_\lambda(\lambda_c^0)x_0 \\ 0 \end{bmatrix} \neq 0.$$

To show this let

$$F_y(y_0, \omega_0, \lambda_c^0) \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} A_\lambda(\lambda_c^0)x_0 \\ 0 \end{bmatrix}.$$

Write out the first row of this equation and then left multiply by v_0^T, which satisfies $v_0^T(A(\lambda_0) - \theta_0 B(\omega_0)) = 0$, to give $q = \theta_\lambda(\omega_0, \lambda_0)$. Thus the inequality reduces to $\mathrm{Re}\,(\theta_\lambda(\omega_0, \lambda_0)) \neq 0$ which is true by assumption. Thus it follows that $G_{y,\lambda}(y_0, \omega_0, \lambda_0)$ is nonsingular.

The result follows by the Implicit Function Theorem.

(ii) Part (*ii*) follows by putting $\lambda = \lambda_c(\omega)$ into (6.5) and taking the real part, noting that $\mathrm{Re}\ \theta_\lambda^0 \neq 0$ by assumption.

□

Figure 6-4 gives a schematic interpretation of the result of Theorem 6.4. Note that if $\theta \in \mathbb{R}$, then the result of Theorem 6.4 reduces to $\lambda_c(\omega) = \lambda_c(0)$ as expected.

Application of Theorem 6.4 to our Stokes preconditioned eigenvalue problem gives the following result.

Corollary 6.5 *At $\lambda = \lambda_c(\Delta t)$, let $(x(\Delta t), \theta(\Delta t))$ be an algebraically simple eigenpair of $(I - \Delta t L)^{-1}(L + N(\lambda_c(\Delta t)))$ with* $\mathrm{Re}\ \theta(\Delta t) = 0$. *Then there is a value of λ, say $\lambda = \lambda_c(0)$, such that $L + N(\lambda_c(0))$ has an algebraically simple eigenvalue $\theta(\lambda_c(0))$ with* $\mathrm{Re}\,(\theta(\lambda_c(0)))$. *Moreover $\lambda_c(0) - \widehat{\lambda}_c(0) = \mathcal{O}(\Delta t^2)$ where*

$$\widehat{\lambda}_c(0) = \lambda_c(\Delta t) - \Delta t \frac{\mathrm{Re}\left\{\theta(\Delta t)\frac{v(\Delta t)^T L x(\Delta t)}{v(\Delta t)^T(I-\Delta t L)x(\Delta t)}\right\}}{\mathrm{Re}\left\{\frac{v(\Delta t)^T N_\lambda(\lambda_c(\Delta t))x(\Delta t)}{v(\Delta t)^T(I-\Delta t L)x(\Delta t)}\right\}}. \tag{6.7}$$

The ability to compute an approximation to $\lambda_c(0)$ given $\lambda_c(\Delta t)$ allows us to detect a Hopf bifurcation. We propose the following approach:

Continue along a path of steady states (using, for example, some predictor-corrector scheme) and

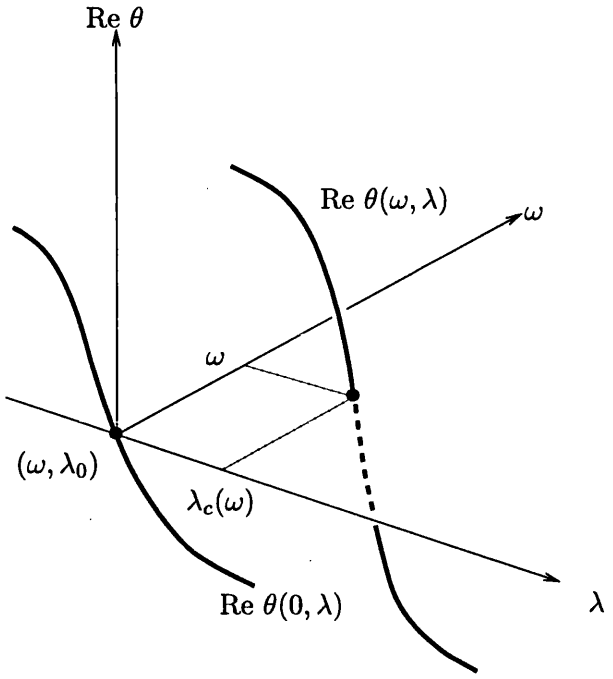


Figure 6-4: Schematic illustrating the result of Theorem 6.4 (i).

(i) compute the the $\lambda_c(\Delta t)$ at which the rightmost eigenvalues of $(I - \Delta t L)^{-1}(L + N_U(U, \lambda))$ have zero real part.

(ii) compute the approximation $\widehat{\lambda}_c(0)$ to $\lambda_c(0)$.

6.4.1 Numerical results for Approach 2

We now give some numerical results for Approach 2 which show how well $\widehat{\lambda}_c(0)$ approximates $\lambda_c(0)$ for different values of Δt

The Tubular Reactor problem Recall the Tubular Reactor problem introduced in Section 6.3.1. We now apply Approach 2 to this problem. Here we approximate $N_{U\lambda}(U, \lambda_c(\Delta t))$ by

$$\frac{N_{U\lambda}(U, \lambda_c(\Delta t) - N_{U\lambda}(U, \lambda_c(\Delta t) - \Delta\lambda))}{\Delta\lambda}$$

for $\Delta\lambda = 1 \times 10^{-6}$.

Table 6.2 shows the critical value $\lambda_c(\Delta t)$ of λ (in this example the Damköhler

Δt	$\lambda_c(\Delta t)$	$\widehat{\lambda}_c(0)$	$\lvert\widehat{\lambda}_c(0) - \lambda_c(0)\rvert$
0.1	0.2637104	0.2614663	2.389×10^{-4}
0.05	0.2625333	0.2612974	6.997×10^{-5}
0.025	0.2618986	0.2612464	1.897×10^{-5}
0.0125	0.2615678	0.2612323	4.882×10^{-6}
0.00625	0.2613988	0.2612286	1.182×10^{-6}

Table 6.2: The critical values $\lambda_c(\Delta t)$ and the error in $\widehat{\lambda}_c(0)$ for varying Δt. Note that $\lambda_c(0) = 0.2612274$.

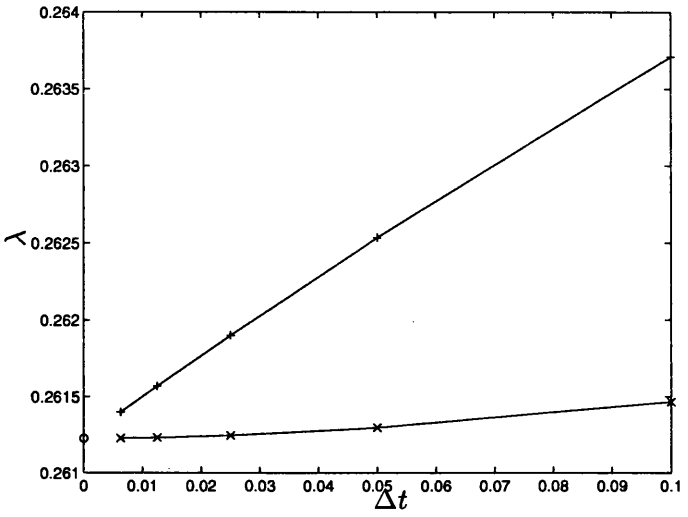


Figure 6-5: $\lambda_c(\Delta t)$ plotted against Δt (+), and the approximation $\widehat{\lambda}_c(0)$ to $\lambda_c(0)$ obtained at Δt against Δt ($\times$). $\lambda_c(0)$ is also plotted ($\circ$).

number) against Δt. This table also shows the error in $\widehat{\lambda}_c(0)$ computed from the eigenvalues of $(I - \Delta t L)^{-1}(L + N_U(U, \lambda_c(\Delta t))$ against Δt. It is clear that $\lambda_c(\Delta t) - \lambda_c(0)$ is of order Δt, and that $\widehat{\lambda}_c(0) - \lambda_c(0)$ is of order Δt^2. This agrees with the analysis of the previous section. Figure 6-5 shows the critical values $\lambda_c(\Delta t)$ and the approximations $\widehat{\lambda}_c(0)$ made from them against Δt.

6.5 Summary

We have shown that the eigenvalues of the Stokes preconditioned Jacobian may be corrected to give second order approximations to the eigenvalues of the Jacobian itself. This leads to an *eigenvalue correction* technique for detecting Hopf bifurcation which

monitors the computed approximations to the eigenvalues of the Jacobian.

The critical parameter value at which the Stokes preconditioned Jacobian has pure imaginary rightmost eigenvalues has been shown to be a distance of order ω from the Hopf bifurcation point. With this result we develop an alternative to *eigenvalue correction* which computes this critical parameter value, and then corrects. We call this method *bifurcation point correction.* It should be noted that this technique requires only *one* correction to be made to approximate a particular bifurcation point.

These techniques allow the detection of bifurcation points without computing explicitly the eigenvalue of the Jacobian matrix. Both methods have been tested on the Tubular Reactor problem.

This work has been submitted to Notes on Numerical Fluid Analysis.

Appendix A

Important results

We state two important results which we use repeatedly in this thesis.

Theorem A.1 (Implicit Function Theorem [13]) *Suppose that*

(i) X, Y, Z *are Banach spaces,*

(ii) $U \subset X$, $V \subset Y$ *are open sets,*

(iii) $F : U \times V \to Z$ *is continuously differentiable,*

(iv) $(x_0, y_0) \in U \times V$,

(v) $F(x_0, y_0) = 0$ *and* $D_x F(x_0, y_0)$ *has a bounded inverse.*

Then there is a neighbourhood $U_1 \times V_1 \in U \times V$ *of* (x_0, y_0) *and a function* $f : V_1 \to U_1$, $f(y_0) = x_0$, *such that* $F(x, y) = 0$ *for* $(x, y) \in U_1 \times V_1$ *if and only if* $x = f(y)$. *If* $F \in C^k(U \times V, Z)$, $k \geq 1$ *or analytic in a neighbourhood of* (x_0, y_0), *then* $f \in C^k(V_1, X)$ *or is analytic in a neighbourhood of* y_0.

Lemma A.2 (ABCD Lemma (Keller [34])) *Given an* $n \times n$ *real matrix* A, $c, b \in \mathbb{R}^n$, $d \in \mathbb{R}$, *consider the* $(n+1) \times (n+1)$ *bordered matrix*

$$M = \begin{pmatrix} A & b \\ c^T & d \end{pmatrix}.$$

(i) If A is nonsingular then M is nonsingular if and only if $d - c^T A^{-1} b \neq 0$.

(ii) If $\operatorname{rank}(A) = n - 1$, *$M$ is nonsingular if and only if*

$$\psi^T b \neq 0 \text{ for all } \psi \in \ker(A^T) \setminus \{0\},$$

and

$$c^T \phi \neq 0 \text{ for all } \phi \in \ker(A) \setminus \{0\}.$$

(iii) If $\operatorname{rank}(A) \leq n - 2$ *then M is singular.*

Bibliography

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK user's guide.* SIAM, 1992.

[2] O. Axelsson and G. Lindskog. On the rate of convergence of the preconditioned Conjugate Gradient method. *Numer. Math.*, 48:499–523, 1986.

[3] Z. Bai, D. Day, J. Demmel, and J. Dongarra. A test matrix collection for non-Hermitian eigenvalue problems. Available from the Matrix Market: `http://www.nist.gov/Matrix Market`, October 1996.

[4] D. Barkley and L. S. Tuckerman. Stokes preconditioning for the inverse power method. Labo. d'Informatique pour la Mecanique et les sciences de l'Ingenieur, Orsay, France and University of Warwick.

[5] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. Templates for the solution of linear systems: building blocks for iterative methods.

[6] A. Booten and H. A. Van der Vorst. Cracking Large-Scale Eigenvalue problems, part I: Algorithms. *Computers In Physics*, 10(3):239–242, May/June 1996.

[7] A. Booten and H. A. Van der Vorst. Cracking Large-Scale Eigenvalue problems, part II: Implementations. *Computers In Physics*, 10(4):331–334, July/August 1996.

[8] A. Booten, D. Fokkema, G. Sleijpen, and H. Van der Vorst. Jacobi-Davidson methods for generalized MHD-eigenvalue problems. Technical report, Mathematical Institute, University of Utrecht, P.O. Box 80010, 3508 TA Utrecht, the Netherlands, 1995.

[9] J.G.L Booten, H. A. Van der Vorst, P.M. Meijer, and H. J. J te Riele. A preconditioned Jacobi-Davidson method for solving large generalized eigenvalue problems. Technical report, Department of Numerical Mathematics, CWI, Amsterdam, The Netherlands, 1994.

[10] C. G. Brooking. *Iterative solution of nonsymmetric linear systems arising from process modelling applications.* PhD thesis, University of Bath, 1996.

[11] P. N. Brown and H. F. Walker. GMRES on nearly singular systems. *SIAM J. Matrix. Anal. Appl.*, 18(1):37–51, January 1997.

[12] F. Chatelin. *Eigenvalues of Matrices.* Wiley, 1993.

[13] S. N. Chow and J. K. Hale. *Methods of bifurcation theory.* Springer-Verlag, 1982.

[14] M. Crouzeix, B. Philippe, and M. Sadkane. The Davidson method. *SIAM J. Sci. Comput.*, 15(1):62–76, Jan 1994.

[15] B. D. Davidson. Large-scale continuation and numerical bifurcation for partial differential equations. *Siam J. Numer. Anal.*, 34(5):2008–2027, October 1997.

[16] E. R. Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *J. Comput. Phys.*, 17:87–84, 1975.

[17] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19(2):400–408, April 1982.

[18] H. A. Van der Vorst and G. H. Golub. 150 years old and still alive: eigenproblems. Universiteit Utrecht Preprint nr 981, October 981.

[19] I. S. Duff, N. I. M. Gould, J. K. Reid, and J. A. Scott. The factorization of sparse symmetric indefinite matrices. *IMA J. of Num. An.*, 11:181–204, 1991.

[20] I. S. Duff, R. G. Grimes, and J. G. Lewis. Sparse matrix test problems. *ACM Trans. Math. Software*, 15(1):1–14, March 1989.

[21] I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Trans. Math. Software*, 9(3):302–325, September 1983.

[22] T. Ericsson. A generalized eigenvalue problem and the Lanczos algorithm. In J. Cullum and R.A. Willoughby, editors, *Large Scale Eigenvalue Problems.* Elsevier Science Publishers B.V., 1986.

[23] T. Ericsson and A. Ruhe. The spectral transformation Lanczos Method for the numerical solution of large sparse generalized eigenvalue problems. *Mathematics of Computation*, 35(152):1251–1268, October 1980.

[24] D. R. Fokkema, G. L.G. Sleipjen, and H. A. Van der Vorst. Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils. Technical report, Department of Mathematics, Utrecht University, 1996.

[25] J. G. F. Francis. The QR transformation—part 2. *Comp J.*, 4:332–345, 1961.

[26] J. G. F. Francis. The QR transformation: a unitary analogue to the LR transformation—part 1. *Comp J.*, 4, 1961.

[27] R. W. Freund. Quasi-kernel polynomials and their use in non-Hermitian matrix iterations. *J. Comp. and Appl. Math.*, pages 135–158, 1992.

[28] T. J. Garratt. *The Numerical Detection of Hopf Bifurcations in large systems arising in fluid mechanics.* PhD thesis, University of Bath, 1991.

[29] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, 1983.

[30] W. Hackbusch. *Iterative solution of large sparse systems of equations.* Springer-Verlag, 1994.

[31] R. F. Heinemann and A. B. Poore. Multiplicity, stability, and oscillatory dynamics of the tubular reactor. *Chemical Engineering Science*, 36:1411–1419, 1981.

[32] M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, December 1952.

[33] A. S. Householder. *The Theory of Matrices in Numerical Analysis.* Blaisdell Publishing Company, 1964.

[34] H. B. Keller. *Applications of bifurcation theory*, chapter Numerical solution of bifurcation and nonlinear eigenvalue problems. Academic Press, 1977.

[35] V. N. Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *USSR Comp. Math. Phys.*, 1(4):555–570, 1961.

[36] C. Lanczos. Solutions of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards*, 49(6):33–53, December 1952.

[37] C. K. Mamun and L. S. Tuckerman. Asymmetry and Hopf bifurcation in spherical couette flow. *Phys. Fluids*, 7(1), January 1995.

[38] The Math Works Inc, Cochituate Place, 24 Prime Way Park, Natick, Mass. 01760. *Matlab Reference Guide.*

[39] K. Meerbergen. *Robust Methods for the calculation of rightmost eigenvalues of nonsymmetric eigenvalue problems.* PhD thesis, Katholieke Universiteit Leuven, 1996.

[40] K. Meerbergen and A. Spence. Implicitly Restarted Arnoldi with purification for the shift-invert transformation. School of Mathematics Preprint, University of Bath.

[41] C. B. Moler and G. W. Stewart. An algorithm for generalized matrix eigenvalue problems. *SIAM J. Numer. Anal.*, 10(2):241–256, April 1973.

[42] R. B. Morgan. Generalizations of Davidson's Method for computing eigenvalues of large nonsymmetric matrices. *Journal of Computational Physics*, 101:287–291, 1992.

[43] R. B. Morgan and D. S. Scott. Generalizations of Davidson's Method for computing eigenvalues of sparse symmetric matrices. *SIAM J. Sci. Stat. Comput*, 7:817–825, 1986.

[44] R. B. Morgan and D. S. Scott. Preconditioning the Lanczos algorithm for sparse symmetric eigenvalue problems. *SIAM J. Sci. Comput.*, 14(3):585–593, May 1993.

[45] B. Nour-Omid, B. N. Parlett, T. Ericsson, and P. S. Jensen. How to implement the spectral transformation. *Mathematics of Computation*, 48(178):663–673, April 1987.

[46] J. Olsen, P. Jørgensen, and J. Simons. Passing the one-billion limit in full configuration-interaction (FCI) calculations. *Chemical Physics Letters*, 169(6):463–472, June 1990.

[47] A. M. Ostrowski. On the convergence of the Rayleigh Quotient Iteration for the computation of the characteristic roots and vectors. i. *Arch. Rational Mech. Anal.*, 1:233–241, 1958.

[48] A. M. Ostrowski. On the convergence of the Rayleigh Quotient Iteration for the computation of the characteristic roots and vectors. ii. *Arch. Rational Mech. Anal.*, 2:423–428, 1959.

[49] B. N. Parlett. The Rayleigh Quotient Iteration and some generalizations for nonnormal matrices. *Mathematics of Computation*, 28(127):679–693, July 1974.

[50] B. N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice Hall, 1980.

[51] U. Rüde and W. Schmid. Inverse multigrid correction for generalized eigenvalue computations. Universität München and Universität Augsburg.

[52] A. Ruhe. The Rational Krylov algorithm for nonsymmetric eigenvalue problems. III: Complex shifts for real matrices. *BIT*, 34:165–176, 1994.

[53] A. Ruhe. Rational Krylov, a practical algorithm for large sparse nonsymmetric matrix pencils. Technical report, Computer Science Division, University of California, Berkely, 1995.

[54] Y. Saad. Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices. *Linear Algebra Appl.*, 34:269–295, 1980.

[55] Y. Saad. SPARSKIT: a basic tool for sparse matrix computations. Technical Report 90-20, University of Minnesota, 1990.

[56] Y. Saad. *Numerical Methods for Large Eigenvalue Problems.* Halsted Press, 1992.

[57] Y. Saad. Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems. *Mathematics of Computation*, 42(166):567–588, April 1994.

[58] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS publishing company, 1996.

[59] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, July 1986.

[60] M. Sadkane. Block Arnoldi and Davidson methods for unsymmetric large eigenvalue problems. *Numer. Math.*, 64:195–211, 1993.

[61] G. M. Schroff and H. B. Keller. Stabilization of unstable procedures: the recursive projection method. *SIAM J. Numer. Anal.*, 30(4):1099–1120, 1993.

[62] D. S. Scott. The advantages of inverted operators in Rayleigh-Ritz approxiations. *Siam J. Sci. Stat. Comput.*, 3(1):68–75, March 1982.

[63] D. S. Scott. Computing a few eigenvalues and eigenvectors of a symmetric band matrix. *SIAM J. Sci. Stat. Comput.*, 3:658–666, September 1984.

[64] G. L. G. Sleijpen, A. G. L. Booten, D. R. Fokkema, and H. A. Van der Vorst. Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems. *BIT*, 36(3):595–633, 1996.

[65] G. L. G. Sleijpen and H. A. Van der Vorst. The Jacobi-Davidson method for eigenvalue problems and its relation with accelerated inexact Newton schemes. Technical report, Mathematical Institute, Utrecht University, Budapestlaan 6, Utrecht, the Netherlands, 1995.

[66] G. L. G. Sleijpen and H. A. Van der Vorst. A Jacobi-Davidson iteration for linear eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 17(2):401–425, April 1996.

[67] D. C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix. Anal. Appl.*, 13(1):357–385, January 1992.

[68] D. C. Sorensen. Implicitly Restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations. Technical report, Department of Computational and Applies Mathematics, Rice University, P.O. Box 1892, Houston, TX 77251, 1995.

[69] A. Stathopoulos, Y. Saad, and C. F. Fischer. Robust preconditioning of large, sparse, symmetric eigenvalue problems. *Journal of computational and applied mathematics*, 64:197–215, 1995.

[70] A. Stathopoulos, Y. Saad, and K. Wu. Dynamic think restarting of the Davidson and the Implicitly Restarted Arnoldi methods. *SIAM J. Sci. Comput.*, 19(1):227–245, January 1998.

[71] G. W. Stewart. Simultaneous Iteration for computing invariant subspaces of non-hermitian matrices. *Numer. Math.*, 25:123–136, 1976.

[72] G. W. Stewart and J. Sun. *Matrix Perturbation Theory.* Academic Press Inc, 1990.

[73] D. B. Szyld. Criteria for combining Inverse and Rayleigh Quotient Iteration. *SIAM J. Numer. Anal.*, 26(6):1369–1375, 1988.

[74] L. N. Trefethen and D. Bau. *Numerical Linear Algebra.* SIAM, 1997.

[75] J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, 1965.